UNIVERSITY OF GENOVA

PHD PROGRAM IN BIOENGINEERING AND ROBOTICS

Planning and Control Strategies for Motion and Interaction of the Humanoid Robot COMAN+

by

Francesco Ruscelli

Thesis submitted for the degree of *Doctor of Philosophy* (33° cycle)

February 2021

Nikolaos G. Tsagarakis	Supervisor
Giorgio Cannata	Head of the PhD program

Thesis Jury:

Joohyung Kim, *University of Illinois*	External examiner
Barkan Uğurlu, *Özyeğin University*	External examiner
Serena Ivaldi, *INRIA*	External examiner

Istituto Italiano di Tecnologia (*HHCM* lab) and
Department of Informatics, Bioengineering, Robotics and Systems Engineering

I would like to thank my parents, who supported me through this Ph.D. journey. I would like to express my gratitude to my supervisor Nikos Tsagarakis, who gave me the opportunity to work on this project and to live this important experience. A huge thank you goes to my colleagues, who always assisted me, especially Dr. Enrico Mingo Hoffman, Dr. Arturo Laurenzi and Dr. Matteo Parigi Polverini. Last but not least, a heartfelt thank you goes to all the friends with whom I shared great memories during these three years.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Francesco Ruscelli
April 2021

Abstract

Despite the majority of robotic platforms are still confined in controlled environments such as factories, thanks to the ever-increasing level of autonomy and the progress on human-robot interaction, robots are starting to be employed for different operations, expanding their focus from uniquely industrial to more diversified scenarios. Humanoid research seeks to obtain the versatility and dexterity of robots capable of mimicking human motion in any environment. With the aim of operating side-to-side with humans, they should be able to carry out complex tasks without posing a threat during operations. In this regard, locomotion, physical interaction with the environment and safety are three essential skills to develop for a biped. Concerning the higher behavioural level of a humanoid, this thesis addresses both ad-hoc movements generated for specific physical interaction tasks and cyclic movements for locomotion. While belonging to the same category and sharing some of the theoretical obstacles, these actions require different approaches: a general high-level task is composed of specific movements that depend on the environment and the nature of the task itself, while regular locomotion involves the generation of periodic trajectories of the limbs. Separate planning and control architectures targeting these aspects of biped motion are designed and developed both from a theoretical and a practical standpoint, demonstrating their efficacy on the new humanoid robot COMAN+, built at Istituto Italiano di Tecnologia. The problem of interaction has been tackled by mimicking the intrinsic elasticity of human muscles, integrating active compliant controllers. However, while state-of-the-art robots may be endowed with compliant architectures, not many can withstand potential system failures that could compromise the safety of a human interacting with the robot. This thesis proposes an implementation of such low-level controller that guarantees a fail-safe behaviour, removing the threat that a humanoid robot could pose if a system failure occurred.

Table of contents

List of figures

Nomenclature

Acronyms / Abbreviations

CD	Centroidal Dynamics
CMM	Centroidal Momentum Matrix
CoM	Center of Mass
CoP	Center of Pressure
CP	Capture Point
CS	Centroidal Statics
CWC	Contact Wrench Cone
DoF	Degree of Freedom
DSP	Digital Signal Processor
HIK	Hierarchical Inverse Kinematics
IK	Inverse Kinematics
KKT	Karush-Kuhn-Tucker
LIPM	Linear Inverted Pendulum Model
LTI	Linear Time-Invariant
MPC	Model Predictive Control
NLP	Non Linear Programming
PC	Preview Control

PTC Pure Torque Control

QP Quadratic Programming

RLS Recursive Least Square

RNEA Recursive Newton-Euler Algorithm

RRT Rapidly-exploring Random Tree

SCIC Semi-Centralized Impedance Controller

SoT Stack of Task

SRDF Semantic Robot Description Format

SVD Singular Value Decomposition

URDF Unified Robot Description Format

VC Virtual Constraint

WPG Walking Pattern Generator

ZMP Zero Moment Point

Chapter 1

Introduction

Robotics studies and develops mechanisms that can be deployed to assist humans: the first robots were used only in controlled environments on an industrial level, but the technological progression fostered the transition to more unstructured and variegated scenarios, helping humans in many aspects of life. Over time, thanks to the ever-increasing level of autonomy and the progress of human-robot interaction, robots were acknowledged not only to be a great resource to replace manpower in repetitive and strenuous tasks or access hazardous locations to avoid life-threatening risks, but employed and deemed useful in a huge variety of situations, to the point that a well-defined categorization is not straightforward anymore.

Surgery, telepresence, education, inspection, rehabilitation, military, exploration are just a few of the emerging applications of robotics. The vast majority of robots are designed according to the specific task to accomplish, therefore each type of robot has unique features that reflect its specific functionality: SCARA, Delta, Gantry robots are specialized industrial robots, drones and self/driving cars are vehicles for transportation, rovers for exploration and inspection. Among many geometries, the category of bio-inspired robots distinguished itself with the enticing premises of versatility. The well-known anthropomorphic arm demonstrated how a certain degree of task-flexibility is often preferred even in industrial scenarios. Fairly recent advancement in technology including compliant actuation, torque control, advanced control algorithms and high-performance mechanical structures allowed the development of animal-shaped platforms, whose distinctive features are articulated locomotion and extensive interaction with the environment: endowing a robot with the intelligence to navigate through the environment and the ability to interact with any surroundings would greatly increase its usefulness. Snake robots proved their efficacy for inspection and repair in otherwise inaccessible areas, as well as in archaeological exploration for slithering through narrow passageways of pyramids. Lizard-like and insect-like robots can easily climb vertical obsta-

cles, while winged robots or finned robots can efficiently fly and swim propelling themselves by modifying their whole body. Quadrupedal robots became extremely popular among researchers and consumers for the many application they are suited for as well as for their possibility to be customized for different purposes. Undeniably, towards the concept of resourceful robots that can navigate the environment using different gaits, such as running, jumping, climbing, crawling, swimming, one of the most versatile and capable animal on earth comes to mind, the *human*. Not only it can traverse with agility unknown environments, but it can adapt to overcome obstacles and interact with the environment in countless ways. Furthermore, in a world where natural evolution favoured the human form and artificial environments are mostly designed for humans, the human body is definitely the most efficient and versatile.

The concept of a human-like robot that can mimic human behaviour roots back in classical times, and throughout history many attempts to build automata were made. However, a real development was only possible in the last years, which saw a rapid growth of the number of humanoid platforms for research all around the world. Two main challenges arise when dealing with bipedal humanoid robots: the design of a mechanism that could rival the human body capabilities and the control of said robot to achieve human dexterity. Besides representing an exciting challenge for researchers, creating versatile and dexterous humanoids capable of mimicking human motion in any environment is a long-desired goal that would boost the benefit that robots represent. While the level of autonomy changes depending on the given task, a humanoid should be able to successfully execute it with minimal supervision, both in operations side-to-side with workers, or substituting the human in demanding, burdensome or dangerous tasks.

In the pursuit of advancing the current state of the art of humanoid robotics, the Humanoid and Human-Centered Mechatronics (HHCM) lab at Istituto Italiano di Tecnologia (IIT) has developed COMAN+, a bipedal robot with the size and the weight of an average person. The ergonomic form and the custom-made actuators featuring high torque and power density place it in the high-performance robots category, and the presence of strain gauge sensors in each actuator unit grants high fidelity torque sensing and control, which, paired with its structural sturdiness and intrinsic passive compliance, makes it ideal for harsh physical interaction tasks.

Being COMAN+ a relatively new product at the beginning of this Ph.D., working on this robotic platform presented many obstacles, concerning both the low-level and the higher behavioural level. However, at the same time, it offered many opportunities to work on diverse architectural levels of a full-fledged humanoid. Compared to classical

(a) The *SEA Snake*, developed by the Biorobotics Lab at CMU.

(b) The robot *Spot*, developed by Boston Dynamics.

(c) The humanoid *TALOS*, developed by PAL Robotics.

Figure 1.1 Different body shapes of bio-inspired robots allows different gaits and skills.

industrial robots, controlling free-floating platforms such as bipeds is considerably harder, due to the additional restrictions caused by their inherent under-actuation. Dealing with a highly redundant system that needs to satisfy balancing conditions and it is subject to non-linear constraints caused by contact forces exchanged with the environments is definitely challenging: nonetheless, it is exceptionally rewarding, given the huge variety of tasks that a humanoid can perform and the great potential of such a platform.

1.1 Thesis overview and objectives

The goal of this Ph.D. dwells in the exploration, development and implementation of control strategies to extend the locomotion and interaction capabilities of COMAN+.

The motivation behind this goal is to provide a full-fledged humanoid platform with the abilities that the human counterpart possess. Mimicking the salient characteristics of humans, the main features this thesis focus on are *locomotion* and *environment interaction*. An inherent requirement to enable these capabilities is elasticity, which allows shock absorption and *compliance* with the surroundings. Finally, while compliance already guarantees a certain

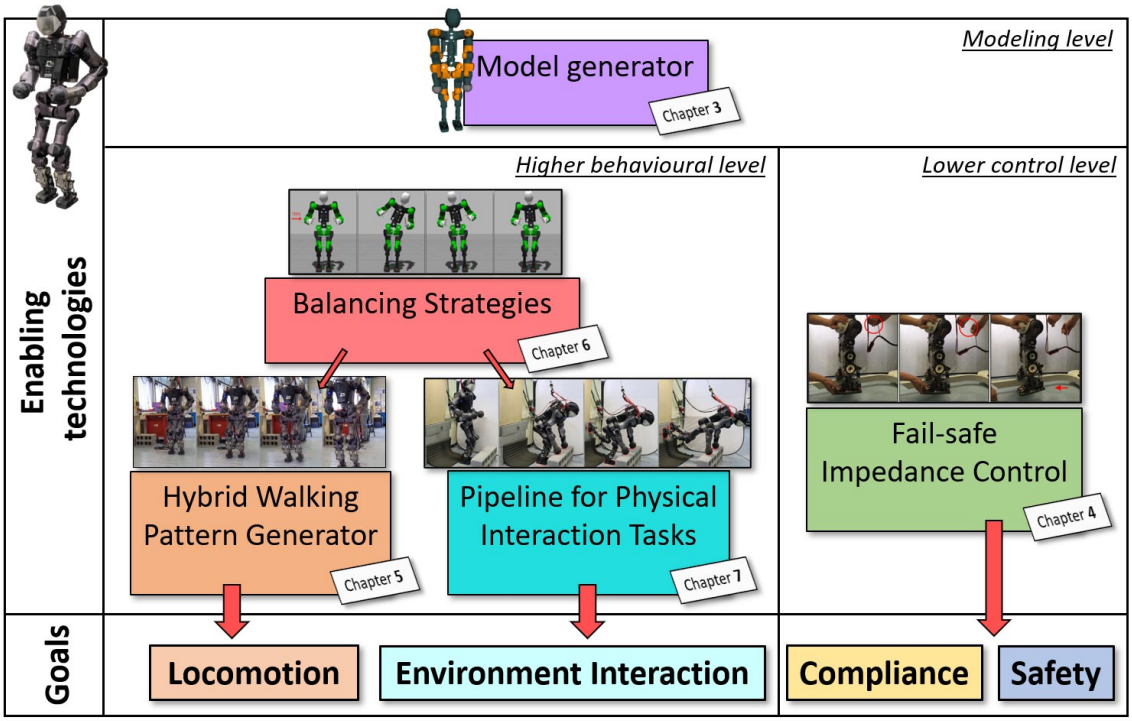


Figure 1.2 Thesis structure with overview of the Ph.D. objectives and the strategies adopted to achieve them.

degree of protection between the robot and the external environment, a critical prerequisite for a bipedal platform moving and interacting is *safety*.

In practice, developing a robot that possesses human-like motion capabilities presents many challenges, both on the control level and on the hardware level. The first should endow the robot with the dexterity and intelligence to adapt to complex tasks and react to external stimuli, the second should grant mobility and robustness while being inherently safe.

In addressing these aspects, the objectives of this Ph.D. list as follows:

- Realize gait pattern generators controllers for COMAN+: as one of the most relevant skills owned by bipeds, endowing the humanoid with the ability to locomote is a key objective that includes the synthesis of feasible gaits and their execution on the robot. A robotic platform able to ambulate freely throughout the surroundings greatly improves its effectiveness, especially for tasks in human environments;

- Develop motion planning and multi-contact control strategies for COMAN+: beside locomotion, another essential skill of a biped is the agile and dexterous interaction with the environment: allowing the robot to execute complex physical interaction tasks represents another central objective of this work;

- Equip COMAN+ with impedance regulation skills that are resilient against communication and torque sensor faults: humanoids are expected to experience frequent interactions with the environment, either when locomoting or while carrying out generic interaction tasks. In doing so, they should be reliable in the presence of hardware failures, minimizing the risk of abrupt and unwanted reactions that could create unsafe conditions in human-robot interaction tasks.

To reach these objectives, aimed towards the humanoid robotic platform COMAN+, the thesis leverages on the following enabling technologies and control components:

i. *Modeling Level*: enables to rapidly build model descriptions compatible with the software used for simulation and real-robot experiments, shortening the gap between hardware design and software deployment. The ROS-based model generator is especially useful when the robot model is accessed frequently, e.g. to modify or substitute a component. Furthermore, it simplifies the process of maintaining and dispatching models of hyper-redundant systems characterized by complex kinematic trees.

ii. *Lower level control*: responsible for rejecting minor disturbances, granting compliance and handling coordination while preventing dangerous movements if hardware failures occur. While allowing for adaptation to the environment and partial energy absorption from impacts, it reduces the risk of abrupt reactions in the presence of communication interruption between the high-level controller and the local actuation regulators.

iii. *Higher behavioural level*: in charge of motion control, it addresses both locomotion and interaction with the environment, two essential skills for a humanoid robot.

    - *Locomotion*: cyclic motions of the limbs aimed at the displacement from one point to another in space. A walking pattern generator (WPG) prescribes reference joint trajectories for a gait, i.e. a stepping motion that does not violate the balancing constraints of a floating-base robot. An effective WPG should comprise on-line direction and step length modifications to allow omni-directionality. Motivated by observations in the bio-mechanics of the human gait, showing how the dynamic motion of the human walk is mainly generated by the sagittal component of the stepping, a hybrid approach is chosen, combining results from different locomotion theories.
    - *Physical interaction*: generic complex whole-body motions involving multi-contact actions. A suitable architecture should account for the planning and

the control for the execution of said motion while guaranteeing self-balance. Leveraging on the strength of COMAN+, it is chosen a demanding task for the average human being as a goal to prove the effectiveness of the framework, the wall-plank. Being composed of strongly non-linear movements and delicate manoeuvres, the devised pipeline should supervise both the *motion planning* and the *motion control* problems.

Each strategy should be tested in simulation and implemented on a robotic platform, facing real-hardware challenges that are seldom overlooked, to bridge the gap between simulation and real demonstrations: every algorithm should be developed with the intention of deploying it on the humanoid robot, meeting the hardware and software requirements of COMAN+. Figure 1.2 shows an overview of the thesis, highlighting the goals of the Ph.D. and how the work has been divided to tackle the relevant topics.

The remainder of this chapter introduces the robot COMAN+ and frames it in a rapidly growing humanoid research scenario. A brief overview of the history and the evolution of humanoid robotics is given, as the thesis is primarily devoted to a robotic platform of the same category. Chapter 2 serves as the foundation for the control of humanoid robots, presenting the relevant theory for the analysis and control of such platforms. Chapter 3 presents the robot used throughout the Ph.D., COMAN+, both on the hardware and the software level. It includes the contribution (i), presenting RoboGen, a dedicated library for URDF and SRDF generation. Chapter 4 focuses on the design and implementation of the lower-level control (ii): in particular, it presents the conceived fail-safe impedance controller. The higher behavioural level contributions can be found in Chapter 5, which introduces the hybrid walking pattern generator for COMAN+ and chapter 7, which describes a planning and control pipeline for complex physical interaction tasks. Chapter 6 is an overview of the balancing strategies implemented for the robot. Finally, chapter 8 discusses the results and summarizes the contribution of this thesis's work.

1.2 Humanoids history

The ambition of creating machines that resemble humans in form and capabilities is deep-rooted in humankind. The desire of crafting artificial beings always dwelt in tales and mythology, before finding fertile ground in recent technological advancements. Countless evidence can be found throughout history and civilizations. Hephaestus, the Greek god of invention and craftsmanship, was responsible for the creation of many statues imbued

with reason and mind, capable of autonomous motion: Talos, a giant automaton of bronze "programmed" to protect the island of Crete, the Golden Maidens, woman-shaped attendants for his palace, golden and silver lions, watchdogs at the entrance of a palace. Jewish legends describe golems, animated heaps of clay that could move and think, while in ancient Egypt the statues of divinities were believed to have a soul, derived from the god they represented. Over time, the idea of humanoid robots rapidly spread in the collective imagination along with the possibility to concretize it, thanks to the ever-growing technological progress of society. The Al-Jazari's programmable mannequins with a few degrees of freedom, the "jacquemars", automated bell-strikers built in the 14^{th} century, the humanoid knight designed by Leonardo Da Vinci are just a few examples of the many attempts made by engineers and inventors. Literature and engineering frequently inspired one another: the robotic dolls built by the Jaquet-Droz family, that could write or play an instrument, prompted Mary Shelley to write Frankenstein: The Modern Prometheus. Science fictions made humanoids a huge theme around which many stories were born both in movies such as "Metropolis" or "Alien", and in books, such as the popular short story "I, Robot", by Isaac Asimov or the dystopian novel "Do Androids Dream of Electric sheep", by Philip K. Dick.

However, it was only until recently that the research community focused on the design and the making of humanoids. Elektro, a robot built by the Westinghouse Electric Corporation in 1937, could talk, move his head and arms, as well as smoke cigarettes. But it is only the first draft of its successors: the Waseda University boosted the development of humanoids with the WABOT project in 1967, which produced the WABOT-1, regarded as the first full-fledged humanoid platform.

Following their trail, universities, companies and new-born labs around the world started researching humanoid motion and cognition. An event that marked humanoid research was DARPA Robotics Challenge (DRC) in 2013, whose goal was developing human-supervised robots capable of executing complex tasks for disaster response: a bench-marking event that drew a line for the humanoid state of the art, and an attempt to recreate a future scenario in which robots could move in a human environment, using human tools and equipment and, more in general, performing equal to any human being. While industrial robots excel in one specific task and surpass by far the ability of the average person in that field, general-purpose robots such as humanoids have always been clunky and wobbly, only remotely resembling the abilities of a human. However, the latest achievements from Boston Dynamics demonstrated not only how the scenario envisioned by the DARPA challenge is possible, but also how new-generation robots can outperform humans, showing unprecedented agility and dexterity in running, jumping, and even accomplishing acrobatics drills, such as handstands and backflips:

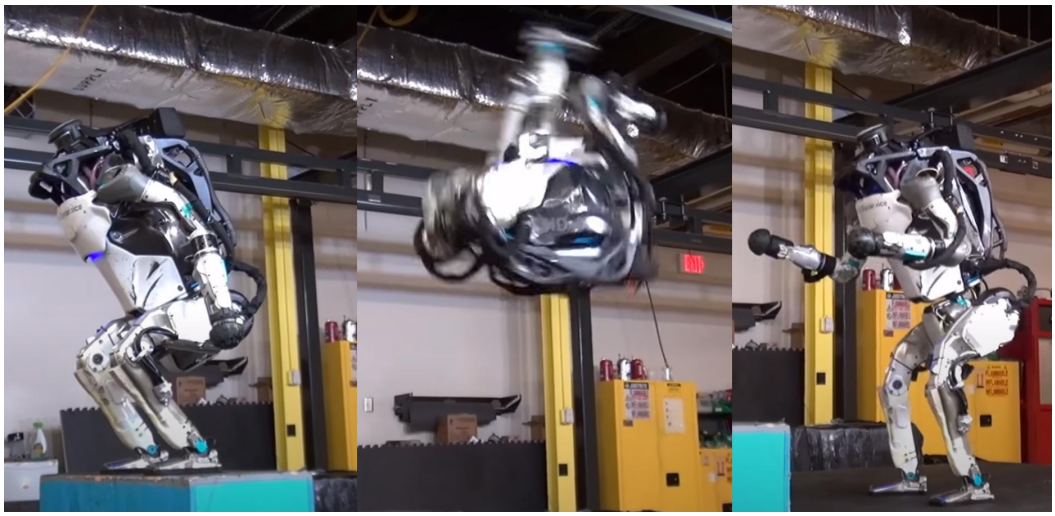


Figure 1.3 The humanoid ATLAS from Boston Dynamics performing a *backflip*. The biped demonstrates remarkable agility and control landing a gymnastic movement that the average human is not capable to execute.

in a surprisingly short amount of time humanoids exceeded any expectation, rivaling humans in terms of fluidity and strength.

1.3 Humanoid robots

Locomotion and interaction with the environment are two essential skills that a general-purpose robot should possess to be autonomous and functional. Among different categories, wheeled and legged robots own higher mobility by far. However, other than reasonably flat terrains on which they are way more efficient, wheeled robots are not agile and versatile as articulated mechanisms. In unstructured grounds and cluttered environments, where the mobility of wheeled robots is heavily compromised, legged robots can adapt to the terrain and exploits their kinematics aptitude to alter their shape to overcome the majority of obstacles while maintaining robust equilibrium. The latter is influenced by the number of limbs that the robot uses to balance: the more supporting limbs, the greater its ability to conserve equilibrium. However, a trade-off between self-balancing and motion versatility has to be made, since quadrupeds can withstand bigger external perturbation, traversing rough terrain with ease, but are less proficient in manipulation and interaction. Bipedal mechanisms are way harder to control due to the relatively small support region and their weight concentrated way above the contacts, but their ability to modulate the body kinematics is superior, allowing

them to adapt the posture according to the work-space, the obstacles in it and the task they are carrying out.

The motion capabilities that a humanoid robot should possess span from a nominal walk to diverse gaits suitable for different circumstances, such as running on cluttered terrains, jumping over obstacles or using any combination of the limbs to interact with the environment, such as climbing or crawling. In light of these capabilities and the fact that the humanoid form is the most compatible with human environments and tasks, the most advantageous shape for a general-purpose robot is the anthropomorphic one. Inspired by biology, the body of a humanoid robot is composed of a stiff skeleton structure, several actuators to simulate human muscles and a wide variety of exteroceptive and proprioceptive sensors to interact with the surroundings. As for the mechanism design challenge, the human body is unrivaled by far, both for its exceptional complexity and the proficiency of its actuators and sensors. Despite the efforts directed towards the development of more human-like parts, such as compliant or tendon based actuators, nimble skeletons and sophisticated sensors, robots are still limited by clunky and rudimentary components compared to their human counterpart. As for the control of humanoids, recent progress enabled impressive achievements in bipedal locomotion and interaction with the environment.

Besides passive walkers and humanoids that only resembled humans in form but not in their structural capabilities (such as many automata, dolls and mannequins built in the past), the technological level in mechatronics required to build a fully powered humanoid was reached less than four decades ago, when Waseda University completed the WABOT-1, the first platform with human proportions that could walk and interact with the environment. A few years later, in 1980, the MIT Leg Lab is founded, a pioneer in the development of extremely capable and dynamic legged robots, the first of which is PETMAN. In 1985 Hitachi Ltd, presents WHL-11, bipedal robot capable only of static walking. The company Honda develops one of the most capable and famous bipeds in the world, ASIMO (Hirai et al., 1998), and its new version in 2011, one of the first autonomous platform to perform a proper run. Before building this final version, Honda experimented from 1985 to 1993 with seven prototypes, from robot E0 (Experimental Model 0) to E6 and continued prototyping until 1997 with other three robots, P1 (Prototype Model 1) through P3 (Hirose and Ogawa, 2007), whose innovative design inspired many other future robots. A successful line of robots was built by the Manufacturing Science and Technology Center (MSTC): the HRP series became famous with the version HRP-2 in 2002, which was used as a test-bed for many promising algorithms. It was followed by the successor HRP-3 and HRP-4 (Kaneko et al., 2008), which demonstrated a very fluid and dynamic motion.

(a) The robot COMAN.

(b) The robot WALK-MAN which participated at the DRC in 2013.

(c) The new robot COMAN+, used in this thesis.

Figure 1.4 The three humanoid robots developed by the HHCM Lab at IIT.

Wabian is a human-size biped walking robot from Waseda University. In 2006, the rising company PAL Robotics built the first autonomous biped humanoid robot in Europe, REEM-A, which was improved with the new version REEM-B (Tellez et al., 2008) and finally REEM-C [1], the first robotic platform based completely on the Robotics Operating System (ROS). The same year, a small humanoid robot developed at Aldebaraan Robotics (now called SoftBanks) called NAO rapidly spread among robotics labs thanks to its open-source code, which the opportunity not only to any user to program it, broaden humanoid research, but also allowed many universities to introduce it as an educational tool. In 2006 another robot became very popular thanks to its catchy design: the child-sized biped iCub (Parmiggiani et al., 2012), built at Istituto Italiano di Tecnologia (IIT) for cognition research. The advent of reliable and compact torque sensors started the successful trend of fully torque controlled robots, such as TORO (TOrque-controlled RObot) developed in 2013 by the German Aerospace Center (DLR) (Englsberger et al., 2014), Samsung's humanoid robot, Roboray (Kim et al., 2012), the most recent release of PAL, TALOS (Stasse et al., 2017), the dexterous and agile ATLAS[2] from Boston Dynamics, and the humanoid robots developed by the HHCM Lab at Istituto Italiano di Tecnologia: COMAN (COmpliant huMANoid) (Tsagarakis et al., 2013, 2011), WALK-MAN (Tsagarakis et al., 2017) and COMAN+. A series of robot worth mentioning is

[1] https://pal-robotics.com/robots/reem-c/
[2] https://www.bostondynamics.com/atlas

the successful family of RABBIT (Chevallereau et al., 2003), MABEL (Grizzle et al., 2009), MARLO (an ATRIAS series robot (Grimes and Hurst, 2012), CASSIE[3] and the brand-new humanoid robot DIGIT[4]. Except for the last one, these robots' design is not completely anthropomorphic, as they are composed only of a lower-body with point feet, they served as prototypes for the flagship DIGIT, a reliable, nimble and rugged humanoid designed to navigate complex environments while carrying packages using a surprisingly dynamic gait. A special mention goes to the world's most dynamic humanoid robot, the previously mentioned ATLAS, which revolutionized the state of the art by showing human-like capabilities such as dynamic walking in extreme conditions, running in rough terrain, jumping on and over obstacles and even performing gymnastic routines and acrobatics moves with unprecedented dexterity and coordination. The main protagonist of this thesis's work is COMAN+, the latest bipedal robot built at IIT.

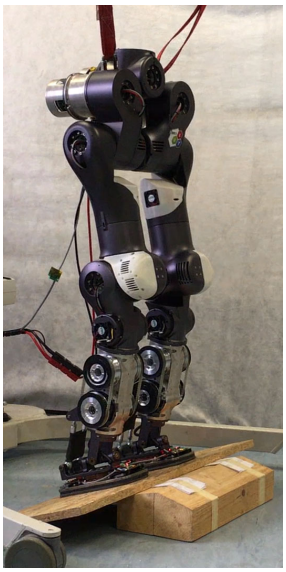
1.4 Contribution

The works collected in this Ph.D. thesis are driven by the desire to endow the robot COMAN+ with locomotion and interaction capabilities, leveraging the features of this robotic platform. The main contributions are the development of different strategies for humanoid motion, both concerning the lower-level control and the higher behavioural level. The work-plan is split between the development of novel strategies and their implementation on the robotic platform COMAN+, facing real-hardware difficulties. The main achievements are divided into three sub-projects, listed in the remainder of this section.

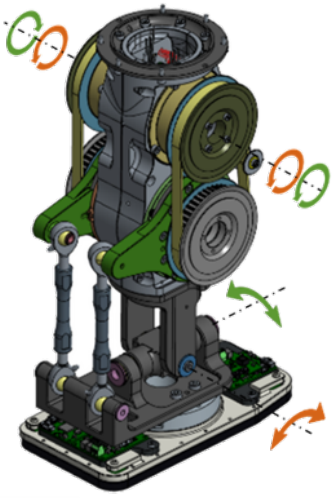
1.4.1 Semi-Centralized Impedance Control

The ability to conform and adapt to external disturbances and impacts is a fundamental prerequisite a humanoid: an effective solution to improve the interaction skills of robotics systems experiencing frequent interactions with the environment is the employment of impedance control schemes. The proposed controller allows adaptations to the environment and partial energy absorption from impacts, mimicking the intrinsic elasticity of human muscles. Additionally, it is immune to instabilities of classical fully centralized methods, that would cause the local joint torque controllers to lose their torque reference inputs. This favors the hardware protection and results in slower and safer robot reactions, especially

[3]https://www.agilityrobotics.com/robots#cassie
[4]https://www.agilityrobotics.com/robots#digit



(a) The legs of the humanoid robot COMAN+. Endowing the robot with active compliance allows it to adapt to different terrains, such as an inclined wooden board.



(b) Close back view of the robot's tibia (CAD) without covers. A dual four-bar mechanism connects the motors with the joint in order to reduce the inertia at the extremity of the leg.

Figure 1.5 The work focused on developing a fail-safe compliant controller for the ankles of COMAN+

important in human-robot interaction. The semi-centralized impedance controller was successfully implemented on COMAN+ for the parallel kinematics mechanism of the ankles. The effectiveness of the method was experimentally proved, performing similarly to a conventional impedance controller. Furthermore, it was experimentally validated in the presence of disconnections: by simulating a hardware failure, it was proved how the ankle does not drift violently, but fails gracefully returning to its nominal position[5].

Contributed paper

F. Ruscelli, A. Laurenzi, E. Mingo Hoffman and N. G. Tsagarakis, "A Fail-Safe Semi-Centralized Impedance Controller: Validation on a Parallel Kinematics Ankle," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018.

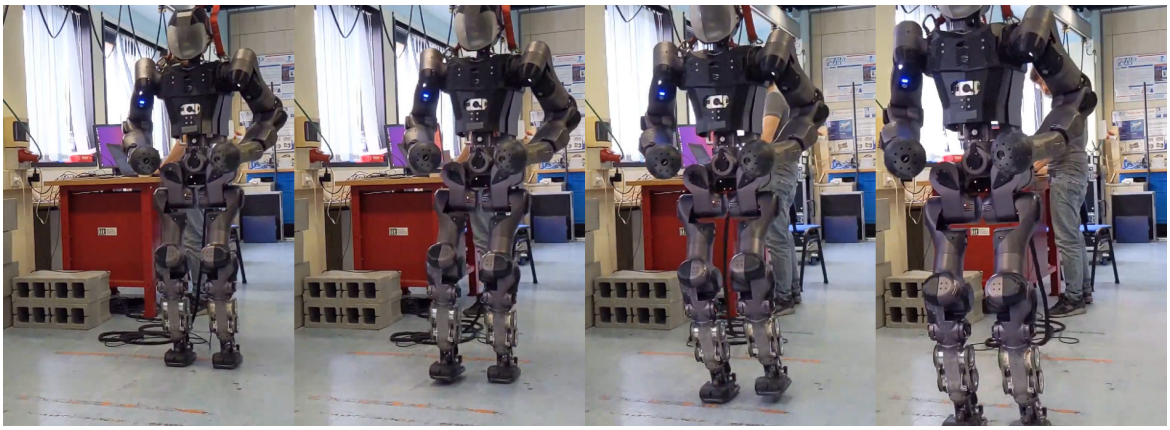


Figure 1.6 COMAN+ walking using the hybrid walking pattern generator.

1.4.2 Hybrid Walking Pattern Generator

The devised strategy allows performing omnidirectional walking without any burdensome parameters tuning: the stepping motion is generated according to a few user inputs such as duration, clearance and stride of the desired step. The selection of the step parameters can be done offline, pre-planning the desired walking trajectory, or modified online, for instance, side-stepping, steering or stepping backward. Virtual Constraints (VC) effectively simplify a complex system such as a humanoid robot into a lower-dimensional system, reducing the degrees of freedom of a robot: the hybrid WPG is lighter in terms of computational complexity than the classical ones and it results in open-loop stable walking of the real robot when no external disturbances are applied. Simulation experiments demonstrate the effectiveness of the strategy: the robot walks and changes the nominal gait according to user-inputs, steering or moving backward.

While the proposed hybrid walking pattern generator (WPG) is promising, because of model inaccuracies, impacts and floor irregularities it is not sufficient for a successful implementation on the real robot. A compliant stabilizer in Zhou et al. (2014) was integrated into the control layer, but the experiments on the real robot didn't meet the expected results, since the robustness of the walking was not improved. This stabilizer only uses the Center of Pressure (CoP) information to adjust the Center of Mass (CoM), hence a more refined strategy for stabilization and disturbance rejection was attempted: the ankle and torso stabilizer. This controller uses the real Capture Point (CP) of the robot, comparing it with the reference CP, hence it requires an observer to measure real COM pose and velocity. However, it was only tried in simulation, since while it showed extremely good results for rejecting external

[5]https://youtu.be/-QULP4xUD9A

pushes when standing still, it is not robust when used for locomotion. Nonetheless, several experiments were performed on COMAN+ with the walking pattern generator, which can walk and steer a few steps before external disturbances and the accumulation of real hardware inaccuracies lead to a loss of balance. Videos of the experiments both in simulation and on the real robot are provided[6].

Contributed paper

F. Ruscelli, A. Laurenzi, E. M. Hoffman and N. G. Tsagarakis, "Synchronizing Virtual Constraints and Preview Controller: a Walking Pattern Generator for the Humanoid Robot COMAN+," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019.

A follow-up paper has been submitted to Frontiers in Robotics and AI, on the research topic "Towards Real-World Deployment of Legged Robots", with the title "Omnidirectional Walking Pattern Generator combining Virtual Constraints and Preview Control for Humanoid Robots". The paper is in the process of being reviewed. The journal article details the full architecture of the omni-directional walking pattern generator, extending the previous paper.

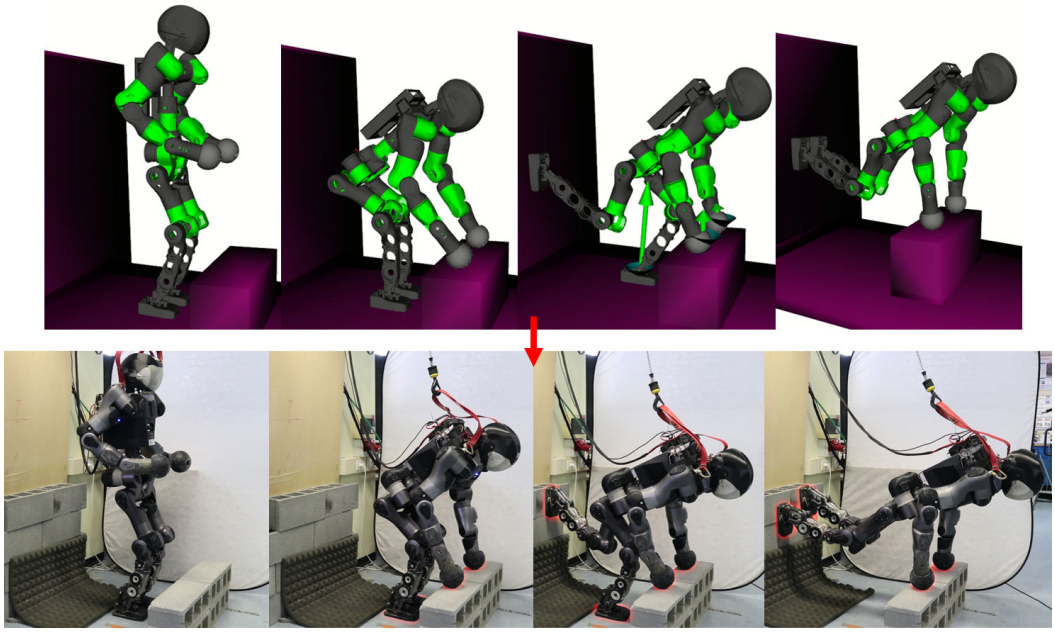


Figure 1.7 Planning and execution of the *wall plank* pose with COMAN+.

[6]https://www.youtube.com/playlist?list=PL7c1ZKncPan7yphDxvZDtaqzmgCRtpT5-

1.4.3 Multi-Contact Motion Planning and Control Framework

This pipeline for planning and control allows a humanoid robot to perform complex real-world physical interaction tasks including highly non-linear motion in narrow spaces and limited by external obstacles and kinematic constraints. The *wall-plank* task is successfully attained with COMAN+: the robot plans and executes a challenging action for the average human being, distributing its weight on the four limbs to keep the body suspended from the ground. The demonstration validates the proposed method, which can robustly produce highly non-linear trajectories in confined spaces. As long as the sequence of maneuvers involves quasi-static motions, the algorithm generates consistent motions for any given physical interaction task, even for robotic platforms with poor flexibility due to their kinematic structure, such as COMAN+. Videos of the experiments show the architecture and the full sequence of motions for the *wall-plank*[7].

Contributed paper

F. Ruscelli, M. P. Polverini, A. Laurenzi, E. M. Hoffman and N. G. Tsagarakis, "A Multi-Contact Motion Planning and Control Strategy for Physical Interaction Tasks Using a Humanoid Robot," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 25-29, 2020, Las Vegas, NV, USA (Virtual).

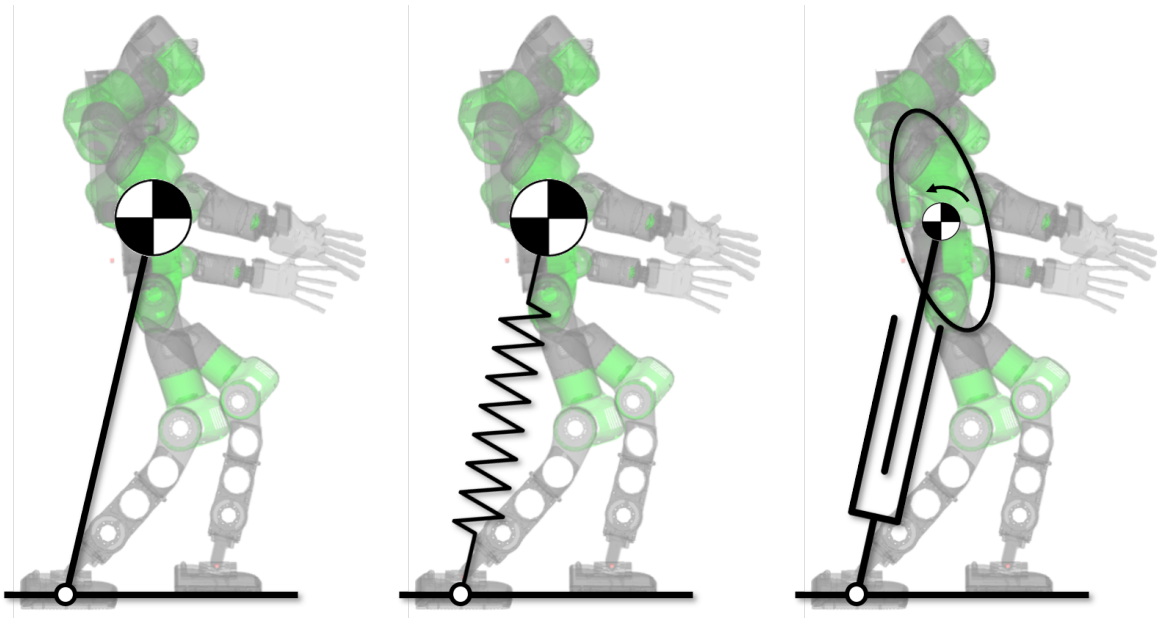
[7]https://www.youtube.com/playlist?list=PL7c1ZKncPan7g0tHDx5jP0kpt5si8G3pR

Chapter 2

Humanoid Motion and Interaction

This chapter furthers the presentation of humanoid robots, drawing parallels between the capabilities of robotic platforms and the humans and goes on summarizing the state-of-the-art theoretical approach used to analyze and control a humanoid robot.

It will become apparent how controlling such robots boils down to handling the interactions with the environment, which corresponds to generating the requisite reaction forces from the contact with the environment. In other words, any motion performed by a humanoid falls back in the macro-category of interaction with the environment constrained by a notion of body equilibrium. Humanoids belong to the category of free-floating base robots. In controlling classical industrial robots, thanks to the fixed-base assumption, the Newton law of action and reaction can be ignored, since any external force exerted on the robot and the inertia generated by the movement of the mechanism are counter-balanced by the ground. The robot is rigidly attached to the ground, and the ground provides the necessary reaction forces to ensure that the base stays anchored to it. Free-floating systems are completely detached from a fixed ground, as the base of the robot floats above the ground from one pose to the other. This property carries many heavy implications that exacerbate the problem of controlling a humanoid robot and introduce under-actuation. While the robot can be modeled as an articulated set of fully actuated rigid bodies, its base is not fixed w.r.t. the ground, additional information is required: three positional and three rotational components totalling the full six degrees of freedom are used to describe the full pose of the base. Since the base is free to move, these degrees of freedom are not actuated, and all the forces and torques on the floating base must be generated by the reaction forces of the links in contact with the environment. Furthermore, not every interaction with the environment is possible: while a fixed-base robot can equally and infinitely "pull" and "push" the ground, depending on the nature of contacts the robot can only induce certain types of forces. For instance, a legged



(a) Linear inverted pendulum model (b) Spring-loaded pendulum model (c) Reaction-mass pendulum with retractable leg.

Figure 2.1 Different approximation of a humanoid robot model.

robot can only press against the ground, and the amount of force exerted tangentially to the ground is bounded by the surface friction.

Indeed, the first obstacle presented by a humanoid is self-balance, i.e. the necessity to continuously satisfy body equilibrium on its limb. Depending on the motion performed, the number of limbs assigned to preserve it varies. The rest of the "free" limbs can be used to improve equilibrium or, more often, execute tasks and manipulate the environment. This raises the second difficulty when controlling bipedal structures: granting effective interaction skills by controlling the unactuated part of the system by generating the required reaction forces with the environment through the contact points, even in the presence of external disturbances.

2.1 Robot Model

In modeling a humanoid robot two extrema are found in literature, both presenting advantages and disadvantages, depending mainly on the mechanism complexity and the objectives to achieve. One models the whole structure of the robot as multi-body system: indeed, when using optimization algorithms to define dynamic motions, this approach produces smooth trajectories and fluid movements, but the full-body model of a humanoid with complex

kinematics performing complex tasks translates into highly non-linear, constrained problems that severely increase the computational burden, making online planning heavy and relatively slow. Oppositely, the other extrema models a humanoid robot as simple dynamical systems that express only the most relevant properties, i.e. the linear inverted pendulum model (LIPM). While this method allows for lighter and faster computations, the main limitation consists in the excessive simplification of the system: since the robot's body is treated as a point-mass, kinematic constraints are not considered and internal body motions (such as rapid movements of the hyper-redundant system to build up useful angular momentum) are overlooked. Since this approach eschews the complex kinematics of a humanoid robot, when used to control actual mechanisms it can produce non-trivial errors that may lead to system failures due to the huge difference between the simple model and the real multi-body robot.

In the middle of these extrema lies a spectrum of different models that, depending on the specific goal or the task to perform, can be selected. Thus, the basic LIPM was progressively modified and many variations were proposed, including a three-dimensional description with the 3-D LIPM (Kajita et al., 2001), adding non-zero angular momentum with the reaction mass pendulum (Lee and Goswami, 2007), including variations of the CoM height (Caron, 2020; Papa and Cappozzo, 1999) or attaching a spring on the leg to simulate the dynamics of running, with the spring-loaded inverted pendulum (Raibert, 1986). These are only a few examples (see Figure 2.1) of versions that emphasize a specific quality that the LIPM fails to describe without recurring to a complete model of the robot, which would require to govern a very complex mechanism. On the other side, the closer the model gets to the real robot's full kinematic structure, the less the control is subjected to inaccuracies of a simple linear model description, and, at the same time, more capabilities of the hyper-redundant robot can be exploited.

To understand the role of the floating base, under-actuation and contact forces, this overview starts by outlining the background of multi-body systems, describing the full model of a humanoid robot and how it requires a different formulation than the model used for classical industrial robot. Subsequently, different strategies are presented which are based on the simplified model of the biped.

To better grasps the nature of a floating base robot, it is convenient to highlight its properties by comparing it with a fixed-base robot. The classical formulation of the Robot Dynamics is briefly presented only to be augmented with the required terms describing the model that best suits a humanoid robot. The next sections briefly introduce the model of a robot and its kinematics and how the same description changes with free-floating assumptions.

2.2 Rigid Body overview

The transformation can be expressed with an homogeneous matrix ${}^{A}\boldsymbol{T}_{B} \in \mathbb{R}^{4\times4}$, which transform a position vector expressed in B to a position vector expressed in A:

$$\boldsymbol{p} = {}^{W}\boldsymbol{R}_{A}^{A}\boldsymbol{p} + {}^{W}\boldsymbol{o}_{A} \tag{2.1}$$

$${}^{A}\boldsymbol{T}_{B} = \begin{bmatrix} {}^{A}\boldsymbol{R}_{B}^{A} & \boldsymbol{o}_{B} \\ \boldsymbol{0}_{1\times3} & \boldsymbol{1} \end{bmatrix} \tag{2.2}$$

The velocity of the rigid body linear $\dot{\boldsymbol{p}}_j$ and angular velocity $\dot{\boldsymbol{\omega}}_j$ as a function of the joint velocities $\dot{\boldsymbol{q}}$:

$$\dot{\boldsymbol{R}} = \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R} \tag{2.3}$$

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_{O} \\ \boldsymbol{\omega} \end{bmatrix} \tag{2.4}$$

$$\boldsymbol{S} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.5}$$

where ${}^{A}\dot{\boldsymbol{\omega}}_{A,B} := \left({}^{A}\dot{\boldsymbol{R}}_{B}^{A}\boldsymbol{R}_{B}^{\top}\right)$ to derive the velocity of the body as the derivative of the homogeneous matrix, ${}^{A}\dot{\boldsymbol{T}}_{B}$.

A different representation of rigid body's motion can be found in *spatial* theory (Featherstone, 2010): intuitively, spatial vectors are 6D vectors that hold a description of the angular velocity of a rigid body as a whole and the linear velocity of one point fixed to the body. This quantity provides a complete description of the body's motion and invariant to the location of the coordinate frame, as opposed to the classical notation 2.4 where the reference frame must always be specified.

$$\boldsymbol{v}_{O} = \boldsymbol{v}_{P} + \overrightarrow{OP} \times \boldsymbol{\omega} \tag{2.6}$$

Both the classical and the spatial theories are just a representation of the same physical phenomenon, and it is always possible to switch from screw transformations to homogeneous transformations using a suitable conversion, as they eventually lead to the same results. However, while the classical notation is intuitive, it lacks the concise form of the *spatial* theory. Furthermore, if computational efficiency is a critical requirement, the large number of multiplications by zeros and ones that emerges using homogeneous matrices makes it unsuitable for fast calculations such as forward and inverse kinematics. However, since the

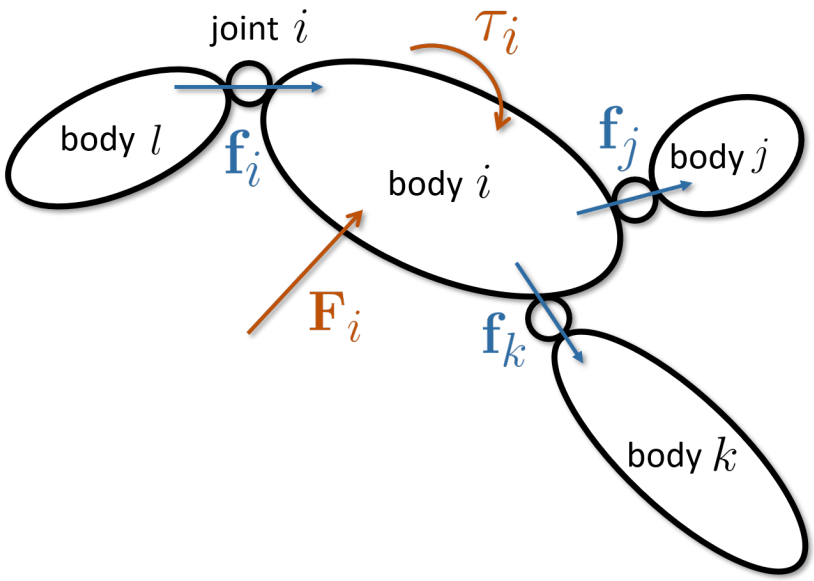


Figure 2.2 An *articulated body* composed of a collection of rigid bodies connected by joints. The mechanism experiences internal forces propagated through the joints and external forces and torques.

twist vector notation is more widespread in robotics and control theory than the spatial vector notation, all the the strategies in this thesis will refer to the former.

2.3 Robot Kinematics

The humanoid robot, from a geometric point of view, is considered as a free-floating multi-body system subject to external forces.

The articulated mechanism is composed of several rigid bodies (called *links*) organized in a tree-like structure. The articulation of the system is provided by the *joints*, which connects the rigid bodies together allowing relative motion of the *child link* w.r.t. the *parent link* along one or more given axis. Each joint provides one or more degrees of freedom (DOFs), whose configuration determines the posture of the robot. In particular, one of the rigid body description illustrated in 2.2 can be exploited to fully define the robot kinematic state. Each link's state is univocally described by the joint variable $\boldsymbol{q}_i$: using this information, together with the type of joint [1], a transformation $T_i(\boldsymbol{q}_J) \in SO^3$ can be defined between a child link $i+1$ and the parent link i to express the position and orientation of the child link w.r.t. the parent link. Clearly, these tools allows to completely describe the mechanism, as the pose of any link can be expressed as a chain of transformations, yielding the forward kinematics

function:

$$f_{\text{fk}}(\boldsymbol{q}) = \boldsymbol{T}_i(\boldsymbol{q}) \tag{2.7}$$

Typically, the forward kinematics consists in expressing the Cartesian pose of an end-effector, which consists in propagating the world frame through the tree structure to the desired link. For instance,

$${}^0\boldsymbol{T}(\boldsymbol{q}_6)_6 = {}^0\boldsymbol{T}(\boldsymbol{q}_1){}^1_1\boldsymbol{T}(\boldsymbol{q}_2){}^2_2\boldsymbol{T}(\boldsymbol{q}_3){}^3_3\boldsymbol{T}(\boldsymbol{q}_4){}^4_4\boldsymbol{T}(\boldsymbol{q}_5){}^5_5\boldsymbol{T}(\boldsymbol{q}_6)_6 \tag{2.8}$$

that allows to map the robot configuration q to the Cartesian pose of the end-effector w.r.t. the world frame.

While the forward kinematics map is a static "snapshot" of the robot describing its configuration at a certain instant of time, the information about the instantaneous motion of the system is conveyed by the *differential kinematics*. Being an articulated mechanism, the motion of each rigid body of the robot is constrained by its structure: hence, a description of the available direction of motion at each configuration is required.

This description can be obtained by differentiating the forward kinematics equations, so as to obtain the linear velocity $\dot{\boldsymbol{p}}$ and the angular velocity $\boldsymbol{\omega}$ of each link as a function of the joint velocities $\dot{\boldsymbol{q}}$. In particular,

$$\dot{\boldsymbol{x}}_i = \begin{bmatrix} \dot{\boldsymbol{p}}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.9}$$

where $J(q) = \frac{df(x)}{dt} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix of the forward kinematics equation. The dimensions of the Jacobian matrix depends on the two vector space involved in the mapping: the Joint space has dimension m, which corresponds to the number of DoF of the robot, while the dimension of a generic Cartesian task is usually $n = 6$ (depending on the task, the number of variables which identify it may be less that six: cleaning a surface, laser wielding..).

Remark The forward kinematics and the differential kinematics are transformations between the Cartesian space and the configuration space of the robot (Joint space). However, it is interesting to notice how, as opposed to the relation between the Cartesian position and joint position, the transformation between the Cartesian velocities and the joint velocities is linear, as the vector $\dot{\boldsymbol{x}}$ is a linear combination of the joint vector $\dot{\boldsymbol{q}}$. In other words, for any desired link, its velocity is a weighted sum of the joint velocities of all the links involved in the motion. Bring the Jacobian matrix is an instantaneous mapping which depends on the

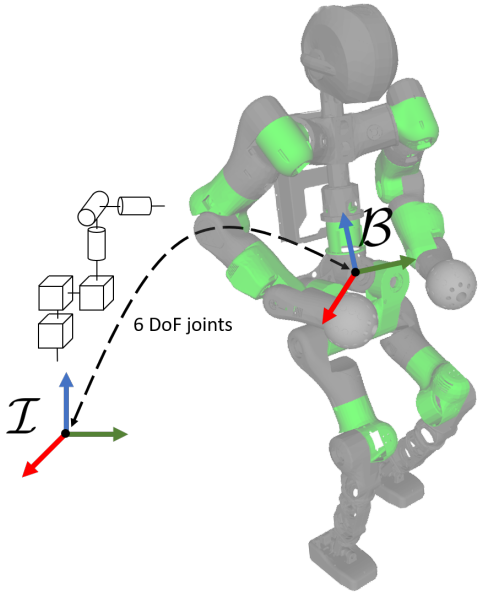


Figure 2.3 Free-floating base robots are not rigidly attached to the ground. They are modeled with a 6 DoF virtual chain connecting the inertial frame $\mathcal{I}$ to the floating base $\mathcal{B}$ fixed to the robot. The joints of the virtual chain, three for the translation and three for the rotation, are not actuated.

joint variables, it is possible to find robot configurations where "directions" in the Cartesian space velocities $\mathbb{R}^6$ do not correspond to any direction in the joint space velocities $\mathbb{R}^n$.

2.3.1 Floating-base Kinematics

For fixed-base robots, the set $\boldsymbol{q}_{\text{joint}} \in \mathbb{R}^n$ is enough to describe all the n rigid bodies of the articulated system in terms of joint variables, since the base link is bolted to the ground and the transformation between the base and the world frame is fixed.

Differently, systems without any link rigidly attached to the world frame are denoted as free-floating. Humanoids fall in this category, as the position of any link w.r.t. the world frame is not fixed. These types of robots require the additional information of where they are located w.r.t. the global frame: a common way is augmenting the state vector of all the actuated joints of the robot with a vector of unactuated variables $\boldsymbol{q}_{\text{base}} \in \mathbb{R}^6$, representing a fictitious chain connecting one link of the robot, denoted as *floating base*, to the world frame.

[1] While a specific design of a humanoid robot or a selected model may include prismatic joints, in this thesis the joint variables will be considered as joint angles, since the majority of legged platforms are composed of revolute joints.

The chain is composed of 6 DoF: a XYZ prismatic joint and a spherical joint, which gives the position and the orientation of the robot's floating base. The full configuration vector is then the concatenation of these two vectors:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{\text{base}} \\ \boldsymbol{q}_{\text{joint}} \end{bmatrix} \in \mathbb{R}^{n+6} \tag{2.10}$$

2.3.2 Inverse Kinematics

The inverse problem of searching for a mapping between a desired Cartesian pose $\boldsymbol{x}$ and the corresponding joint vector $\boldsymbol{q}$ is called the problem of Inverse Kinematics (IK). This essentially comes to inverting the line2.9, to obtain:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}} \tag{2.11}$$

Depending on the properties of the Jacobian, which reflect the configuration of the system, inverting the Jacobian matrix $J(q) \in \mathbb{R}^{m \times n}$ requires different approaches. In the trivial case, the inversion is straightforward as only one solution exists:

1. $\boldsymbol{J}(\boldsymbol{q})$ is square: the DoFs of the robot equals the dimension of the Cartesian task
2. $\boldsymbol{J}(\boldsymbol{q})$ has full rank: motion can be realized along any instantaneous Cartesian direction, i.e. the configuration of the robot is not singular.

For redundant robots such as humanoids, the solution of the system 2.9 is not unique, as the Joint space dimension m is higher than the dimension of the Cartesian task desired, i.e. there exist more equations than unknowns. Hence, a closed-form of the solution does not exist. Intuitively, this means that the same position and orientation of an end-effector in the Cartesian space can be obtained in an infinite number of ways, since the robot has at its disposal more DoF than the required: in other words, there exist an infinite number of robot configurations leading to the same exact pose of the end effector. The problem is then finding a suitable solution among the continuum of available solutions. One viable option to solve this redundant problem is using the Moore-Penrose pseudoinverse:

$$\dot{\boldsymbol{q}}_s = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}} \tag{2.12}$$

which generates the minimum norm solution, i.e. joint velocities $\|\dot{\boldsymbol{q}}_s\|$, and, given $\dot{\boldsymbol{x}}_s = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}_s$, it also gives the minimum norm error $\|\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_s\|$ of the Cartesian velocity. While it could be argued that these properties make it a successful method, one must remember that the

differential kinematics mapping is instantaneous: for this reason, even if the pseudo-inverse guarantee a minimum norm solution, this solution is minimized only instantaneously, and it can become arbitrarily large in the next configuration as the robot approaches a singularity (Baillieul et al., 1984).

A simple way to overcome this drawback is using the damped least-square inverse of the Jacobian matrix $\boldsymbol{A}^{\dagger} \approx \boldsymbol{A}^{\top}\left(\boldsymbol{A}\boldsymbol{A}^{\top} + \lambda \boldsymbol{I}_m\right)^{-1}$, proposed by Wampler (1986). This approach provides a non-singular matrix that can be used to get a solution even in configurations close to the singular one. Naturally, this comes at the cost of a solution which is only approximate: the damping factor λ is a weight of the minimum norm solution $\|\dot{\boldsymbol{q}}_s\|$ w.r.t the minimum task tracking error $\|\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_s\|$ (see Equation 2.11). High values of λ grant robustness to singularities but deteriorate the solution in the neighbourhood of singular configurations. In fact, this term influence blindly all columns of the transformation. A more refined approach is presented in Maciejewski and Klein (1989): by decomposing the matrix $JJ^{\top}$ matrix using the singular value decomposition (SVD), it is possible to analyze which directions are close to a singularity and selectively damp it without influencing all the other directions.

Another drawback of the pseudoinverse is that redundancy cannot be explicitly exploited. Having a redundant structure implies that the robot can self-move without violating a given task, i.e. change its joint configuration while keeping a desired end-effector pose or trajectory. Thus, the robot structure could accomplish additional constraints tasks without influencing the main one.

A different approach that allows tractability of this non-trivial problem is task-oriented control. Building upon the notion of task, this strategy describes a framework to define and prioritize different behaviors of the robot, such as CoM position control, balance control in specific stances, interaction with the environment, obstacle avoidance or end-effector control. Similar to human behavior, which focuses on specific tasks and subconsciously controls the rest of the body to achieve them, this strategy directs the locus of attention to specific regions of the robot, depending on the objective to satisfy.

This task-oriented approach naturally leads to the concept of hierarchy: as the redundant robot can manage multiple tasks simultaneously, one effective solution is assigning a priority to each task and organizing them in a stack where lower priority tasks are satisfied without conflicting with the higher priority tasks.

In particular, task-oriented control exploits high redundancy dividing the configuration space into a task space and null space. Lower priority objectives lie in the kernel of higher priority ones, preventing minor tasks to jeopardize critical behaviors such as body balance or

locomotion. Hence, the kernel is used to impose additional constraints when secondary tasks need to be performed without interfering with core tasks.

A closed-form solution for this is found in Nakamura et al. (1987):

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{x}} + \tilde{\boldsymbol{J}}_y^{\dagger}\left(\dot{\boldsymbol{y}} - \boldsymbol{J}_y\boldsymbol{y}^{\dagger}\dot{\boldsymbol{x}}\right) + \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)\left(\boldsymbol{I} - \tilde{\boldsymbol{J}}_y^{\dagger}\tilde{\boldsymbol{J}}_y\right)\boldsymbol{z} \tag{2.13}$$

where $\tilde{\boldsymbol{J}}_y = \boldsymbol{J}_y\left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)$. This formulation allows to specify a lower priority task y in the null (Walker and Marcus, 1988). Task hierarchy can also be achieved by framing the IK as an optimization problem: this strategy is of particular interest for this thesis, as it is used to set up and solve every IK problem for the humanoid robot. Differently from the closed-form solution described in 2.13, this formulation is used to easily stack any desired number of tasks in a hierarchical fashion. As long as there are "available" DoFs in the system, each task is solved in the null space of the higher priority one. Furthermore, it allows specifying constraints for each task, useful to include joint limits or joint velocities. A generic task at the n^{th} level of priority can be written as:

$$\min_{\boldsymbol{q}_i} \quad \|\boldsymbol{J}_i\boldsymbol{q}_i - b_i\|^2 + \varepsilon\|\boldsymbol{q}_i\|^2 \tag{2.14}$$

$$\text{s.t.:} \quad b_l \leq \boldsymbol{q}_i \leq b_{\mathrm{u}} \tag{2.15}$$

$$\begin{aligned} A_{i-1}\boldsymbol{q}_{i-1}^{*} &= A_{i-1}\boldsymbol{q}_i \\ &\vdots \\ A_0\boldsymbol{q}_0^{*} &= A_0\boldsymbol{q}_i \end{aligned} \tag{2.16}$$

where the higher priority solutions $\boldsymbol{q}_i^{*}, \forall i < n$ are injected in the problem as equality constraints 2.16, so their optimality is not influenced by the n^{th} solution. Notice how a generic solution $\boldsymbol{x}^{*}$ of the pseudo-inverse in (2.12) corresponds to the solution minimizing the least-squares problem (2.14) not subject to any constraints.

2.4 Robot Dynamics

The dynamics model embodies the mathematical relationship between the external forces, the driving torques and the corresponding motion of the robot. Starting from the assumption that the robot is composed of rigid bodies connected in an articulated structure, it is possible to model its dynamics as a second-order system: this corresponds to a linear relationship between the acceleration of the rigid bodies in the robot and the generated torques.

Similarly to kinematics, the "direction" of the dynamic relation can be specified. Direct dynamics is used to compute the motion of the system, i.e. the acceleration of each joint $\ddot{\boldsymbol{q}}(t)$,given the vector of torques $\boldsymbol{\tau}(t)$ applied to the joints and the initial conditions $\boldsymbol{q}(t_0), \dot{\boldsymbol{q}}(t_0)$:

$$\ddot{\boldsymbol{q}} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}) \tag{2.17}$$

Inverse dynamics, on the other hand, yields the joint torques to achieve a desired trajectory $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)$ as a function of the known position, velocity and acceleration of each joint:

$$\boldsymbol{\tau} = f^{-1}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \tag{2.18}$$

The Euler-Lagrange approach is the classical method to retrieve the dynamical model of the robot. This solution is intuitive and starts form the kinetic and potential energies of the robot to extract the closed-form of the model:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.19}$$

Each term can be assigned a physical explanation. In particular, $\boldsymbol{B}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{g}(\boldsymbol{q})$ only depend on the robot configuration: The inertia matrix $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is a positive definite, square matrix:

$$\boldsymbol{B}(\boldsymbol{q}) = \sum_i J_{ti}^\top m_i J_{ti} + J_{Ri}^\top \boldsymbol{I}_i J_{Ri} \tag{2.20}$$

it consists of the moment of inertia at each joint i when all the other joints are fixed and the inertia coupling between two joints, i.e. the effects of the acceleration of a joint on another. The non-linear dynamics effects are grouped in the matrix $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$. which comprise the centrifugal and Coriolis effects due to the joint velocities. The sum of the moment generated by the gravity on each joint, given the robot configuration, is accounted in $\boldsymbol{g}(\boldsymbol{q})$, and finally, the term $\boldsymbol{\tau} \in \mathbb{R}^n$ represents the torques applied to each joint.

While this formulation is useful to analyze the system, it is not computationally efficient. If speed is required, a very fast method exploits spatial algebra to formulate the Recursive Newton-Euler Algorithm (RNEA), which exploits the serial structure of the robot.

2.4.1 Floating-base dynamics

Differently from a fixed-base robot, a floating base system has two critical aspects that need to be taken into account, stemming from the fact that the robot is not rigidly attached to the ground:

- *under-actuation*: the base is free to move in space, introducing additional degrees of freedom that cannot be directly controlled. The robot can only apply torques to its n joints.
- *reaction forces*: any contact with the environment and the relative contact forces will produce equal and opposite reaction forces that are not counterbalanced, as opposed to classical fixed-base configuration where the ground "absorbs" every reaction force.

Given the augmented vector (2.10), torques can only be applied to the last n elements, as the first six element represents the free-floating base. This under-actuation can be injected in the model adding the *actuation* matrix $\boldsymbol{S} = [\boldsymbol{0}, \boldsymbol{I}]$, that maps the actuated torques $\boldsymbol{\tau} \in \mathbb{R}^n$ to the under-actuated model:

$$\boldsymbol{S\tau} = \begin{bmatrix} \boldsymbol{0}_{6\times 6} \\ \boldsymbol{I}_{n\times 6} \end{bmatrix} \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{0}_{6\times 1} \\ \boldsymbol{\tau}_{n\times 1} \end{bmatrix} \tag{2.21}$$

The external forces and the reaction forces generated by the environment against the robot directly influences the linear and rotational acceleration of the center of mass. External shoves are detrimental to the robot, as they interfere with its postural equilibrium, while contact forces stem from the desired interactions with the environment, and are generally to control the robot motions. These forces can be unilateral, as in most of the cases the robot interacts with surfaces that can only be pushed, or bilateral, if the robot can grasp a fixed object to pull it. In any case, contacts are constraints that remove one or more DoF to the robot only temporarily: they are not fixed, as they are realized thanks to the friction between the two touching bodies, and their feasibility depends on the corresponding friction cones. The concepts of under-actuation, reaction forces and contacts with the environment are discussed in the next sections of this chapter.

2.4.2 Task space and Joint space

While the joint space dynamic model establishes a relationship between joint torques and joint accelerations, given a specific task $\boldsymbol{x}(\boldsymbol{q}) \in R^m$ of the robot, such as the pose of an end-effector ($m = 6$), it may be useful to infer the equation connecting it to the forces acting on the mechanism.

The differentiation of the generic task yields a relationship between task and joint accelerations:

$$\frac{d^2}{dt^2}\boldsymbol{x}(\boldsymbol{q}) = \ddot{\boldsymbol{x}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} \tag{2.22}$$

If the Jacobian matrix $\boldsymbol{J}$ is invertible, i.e. in the absence of singularities in the mapping of the given task onto the joint space, the task-space model can be obtained by isolating the acceleration term in the joint space model (2.19) and plugging it in (2.22):

$$\boldsymbol{\Lambda}\ddot{\boldsymbol{x}} = \overline{\boldsymbol{J}}\boldsymbol{\tau} + \boldsymbol{F}_x + \boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{2.23}$$

where $\boldsymbol{\Lambda} = (\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^\top)^{-1} \in \mathbb{R}^{m\times m}$ is the task-space inertia matrix, $\boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the bias force vector (a parallel to the non-linear terms vector $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$. The vector of joint forces $\boldsymbol{\tau}$ is projected onto the task-space by the so-called *dynamically-consistent pseudo-inverse* matrix $\overline{\boldsymbol{J}} = \boldsymbol{B}^{-1}\boldsymbol{J}^\top\boldsymbol{\Lambda}$. Such Jacobian matrix maps the joint-space torques $\tau \in \mathbb{R}^n$ into task-space forces $\boldsymbol{F}_{\tau|\boldsymbol{x}} \in \mathbb{R}^m$. Torques that are dynamically consistent with the task, as introduced in Khatib (1987), are the torques acting in the null space of the matrix, which are not transmitted to the task-space and do not affect it:

$$\boldsymbol{\tau} = \boldsymbol{J}^\top(\boldsymbol{q})\boldsymbol{F}_{\tau|\boldsymbol{x}} + \left(\boldsymbol{I}_n - \boldsymbol{J}^\top\overline{\boldsymbol{J}}^\top\right)\boldsymbol{\tau}_0 \tag{2.24}$$

for any vector of torques τ_0. For redundant robot such as humanoids, the dynamical consistency w.r.t a task consists in decoupling joint-space torques into a force that satisfy the task and an internal force that can be exploited without interfering with the task itself.

2.4.3 Centroidal Dynamics and Nonholonomy

Complex articulated mechanisms such as humanoids are subject to gravity, ground reaction forces, interaction forces from the environment in addition to gravity. Usually, to simplify a mechanical system, the first approach is studying the dynamics of its CoM. Regardless of the system, the CoM encloses many important properties. It is the point where the robot's total mass is concentrated, and consequently where the resultant gravity force acts. Furthermore, it is where the aggregated linear momentum has meaningful definition: as stated in Euler's first law, the linear momentum of a body is equal to the product of the total mass of the body and the velocity of its CoM. In particular, the so called *centroidal momentum* consists in the sum of all the link momenta, after their projection to the robot CoM. It is the vector $\boldsymbol{h} \in \mathbb{R}^6$:

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{L} \end{bmatrix} = \boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{A}_p(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{A}_L(\boldsymbol{q})\dot{\boldsymbol{q}} \end{bmatrix} \tag{2.25}$$

where $\boldsymbol{h}_{\mathrm{L}}$ is the linear momentum and $\boldsymbol{h}_{\mathrm{A}}$ is the angular momentum expressed w.r.t. the CoM. The matrix $\boldsymbol{A}$ is called the *Centroidal Momentum Matrix* (CMM), which defines a linear relationship between the centroidal momentum and the joint velocities.

The importance of the linear momentum is immediately evident, as it can be integrated to obtain the CoM position, since it only depends on the current configuration $\boldsymbol{q}$ of the robot. Unfortunately, the angular momentum has no corresponding meaning: being a non-holonomic equation, it cannot be integrated to retrieve information on the robot orientation.

The rate of change of the centroidal momentum $\dot{\boldsymbol{h}} = \begin{bmatrix} \dot{\boldsymbol{p}}^\top & \dot{\boldsymbol{L}}^\top \end{bmatrix}^\top$ yields the *centroidal dynamics* equation:

$$\frac{d}{dt}\begin{bmatrix} m\dot{\boldsymbol{x}}_{\mathrm{com}} \\ \sum_k (\boldsymbol{x}_k - \boldsymbol{x}_{\mathrm{com}}) \times m_k \dot{\boldsymbol{x}}_k + \boldsymbol{I}_k \boldsymbol{\omega}_k \end{bmatrix} = \begin{bmatrix} \sum_{\mathrm{k}} \boldsymbol{f}_{\mathrm{k}} \\ \sum_{\mathrm{k}} (\boldsymbol{x}_k - \boldsymbol{x}_{\mathrm{com}}) \times \boldsymbol{f}_{\mathrm{k}} + \boldsymbol{\tau}_{\mathrm{k}} \end{bmatrix} \tag{2.26}$$

where $\boldsymbol{f}_{\mathrm{k}}$ and $\boldsymbol{\tau}_{\mathrm{k}}$ are external forces and torques acting on the system. As noted in Wieber (2006), it is worth pointing out that, without any external wrenches, i.e. under the assumption that only control forces act on (2.26), the system conserves its kinetic energy and the left hand side of the equation is equal to zero, and, therefore:

$$m\dot{\boldsymbol{x}}_{\mathrm{CoM}} = \text{constant} \tag{2.27}$$

$$\sum_k (\boldsymbol{x}_k - \boldsymbol{x}_{\mathrm{CoM}}) \times m_k \dot{\boldsymbol{x}}_k + \boldsymbol{I}_k \boldsymbol{\omega}_k = \text{constant} \tag{2.28}$$

The first equation implies that internal forces does not influence the position of the CoM: if it is following a trajectory, it will continue on following it, and if it has zero velocity, the system will remain still, whatever the input torques are. Due to the non-nonholonomic nature of the second equation, a relation between the angular momentum and the body configuration does not exist: the equation constraints the velocities of the bodies $\boldsymbol{x}_k$ and $\boldsymbol{\omega}_k$, without telling anything about their position. Two observation can be made:

- a floating base system such as a humanoid can only move its CoM with the aid of the reactive forces coming from the environment
- it can use internal motions to rearrange the overall orientation of its body without influencing the position of the CoM

2.4.4 Contact Dynamics

Especially for legged robots such as humanoids, standing on the ground with two feet or leaning on a wall with one hand, there exist a subset of reaction forces which corresponds to fixed contacts with surrounding surfaces. Thus, the forward kinematics (2.7) that maps the point in contact with the environment will be constant:

$$f_{fk}(\boldsymbol{q}) = \boldsymbol{x}_c \iff \frac{d}{dt} f_{fk}(\boldsymbol{q}) = 0 \iff \boldsymbol{J}_C \dot{\boldsymbol{q}} = 0 \tag{2.29}$$

where $\boldsymbol{J}_C \in \boldsymbol{R}^{k \times n}$ is the *contact Jacobian* matrix. Due to the nature of the dynamic system, it is possible to obtain a linear relationship between the contact constraint and the robot model by taking a further derivation of equation (2.29). By considering the equation of motion (2.19) augmented with the *actuation* matrix (2.21):

$$\begin{aligned} &\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{S}\boldsymbol{\tau} + \boldsymbol{J}_C^\top \boldsymbol{F}_C \\ &\boldsymbol{J}_C \ddot{\boldsymbol{q}} + \boldsymbol{J}_C \dot{\boldsymbol{q}} = 0 \end{aligned} \tag{2.30}$$

where $\boldsymbol{F}_C \in \mathbb{R}^k$ are the *contact forces* applied on the robot through the k links in contact with the environment. The resulting equations defines a constrained relationship between Cartesian space forces $\boldsymbol{F}_C$, actuation torques τ, and the kinematic accelerations $\ddot{\boldsymbol{q}}$ of the robot.

As long as there are more DoF in the robot than the contact constraints and the contact Jacobian matrix $\boldsymbol{J}_C$ has full rank, from equations (2.30), thanks to the invertibility of the symmetric and positive-definite matrix $M(\boldsymbol{q})$, an equation for the contact forces can be obtained:

$$\boldsymbol{F}_\mathrm{C} = \left[\boldsymbol{J}_\mathrm{C} \boldsymbol{B}^{-1} \boldsymbol{J}_\mathrm{C}^\top\right]^{-1} \boldsymbol{J}_\mathrm{C} \boldsymbol{B}^{-1} (\boldsymbol{h} - \boldsymbol{S}\boldsymbol{\tau}) - \boldsymbol{J}_\mathrm{C} \dot{\boldsymbol{q}} \tag{2.31}$$

It is worth noticing how this equation relate how the torques $\boldsymbol{\tau}$ applied at the joints are mapped into contact forces $\boldsymbol{F}$ applied by the environment at the contact points. As a matter of fact, when the robot is commanded with a vector of joint torques τ, the robot exert a force $-\Phi(\tau)$ on the environment, and in turn the environment exert the opposite force $\Phi(\tau)$ back on the robot, which will directly influence the motion of the floating base.

Remark The robot can only rely on these forces to generate a CoM displacement. However, most of the times these contact forces are limited by unilaterality (the robot can only push, and not pull, the ground)

$$\boldsymbol{n}^\top \boldsymbol{f} \geq 0 \tag{2.32}$$

and friction (especially for non-coplanar contacts, frictions plays an important role for the equilibrium of the whole robot).

2.5 Robot Motion and Interaction Analysis

This section presents the most relevant concepts to analyze the motion of a floating base robot. As they were mostly conceived for studying the motion of legged robots, now that the general background for floating-base and redundant robots are laid, the focus will be directed towards a very specific type of robotic platform, the humanoid.

2.5.1 Static Equilibrium

The simplest type of body equilibrium of an articulated mechanism is known as *static equilibrium* or *static balancing*[2]. Under the assumption that all the contact points lay on the same horizontal surface, it states that a system can remain static as long as the projection of the CoM on that surface lays inside the convex polygon delimited by the contact points of the robot, known as "convex hull" or "support polygon". Although the assumptions for these criteria are tight, the described scenario is very common, as it corresponds to the robot standing still on the ground. The vertical and the horizontal momenta in (2.26) can be decoupled, resulting in:

$$-\sum_k m_k g\boldsymbol{n} + \sum f_{z,k}\boldsymbol{n} = \mathbf{0} \qquad \Longrightarrow \quad \text{vertical component} \tag{2.33}$$

$$-\sum_k \boldsymbol{x}_k \times m_k g\boldsymbol{n} + \sum \boldsymbol{p}_k \times f_{z,k}\boldsymbol{n} = \mathbf{0} \qquad \Longrightarrow \quad \text{horizontal component} \tag{2.34}$$

where $\boldsymbol{p}_k$ is the k^{th} and $f_{z,k}$ is the vertical component applied to it. Compared to the centroidal dynamics equation, the terms $\boldsymbol{f}_{\mathrm{k}}$ and $\boldsymbol{n}_{\mathrm{k}}$ are equal to zero, since no external forces and torques are acting on the system. By dividing the second equation by the first, the static equilibrium criterion is obtained as:

$$\frac{\sum_k m_k \boldsymbol{x}_k}{\sum_k m_k} \times \boldsymbol{n} = \frac{\sum f_{z,k}\boldsymbol{p}_k}{\sum f_{z,k}} \times \boldsymbol{n} \tag{2.35}$$

where the left side of the equation is the cross product between the CoM position $\boldsymbol{x}_{CoM} = \frac{\sum_k m_k \boldsymbol{x}_k}{\sum_k m_k}$ and the normal to the ground $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$, while the right side is a weighted sum of the contact positions $\boldsymbol{p}_k$. Indeed, it corresponds to a point of the contact plane, and the unilateral constraints assumption (2.32) tells that this average is convex, so that the point is inside the convex hull delimited by the contacts. This point is known as the *Center of*

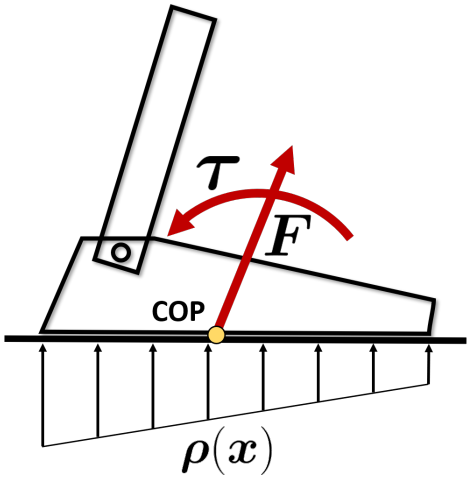


Figure 2.4 The CoP is a local quantity w.r.t two bodies in contact. It is defined as the average point of the contact points weighted by their respective pressures The distribution of reaction forces along the sole $\rho(\boldsymbol{x})$ depends on the gravity-inertial forces and torques applied on the robot. The CoP coincides with the point where the resultant contact force and torques are computed.

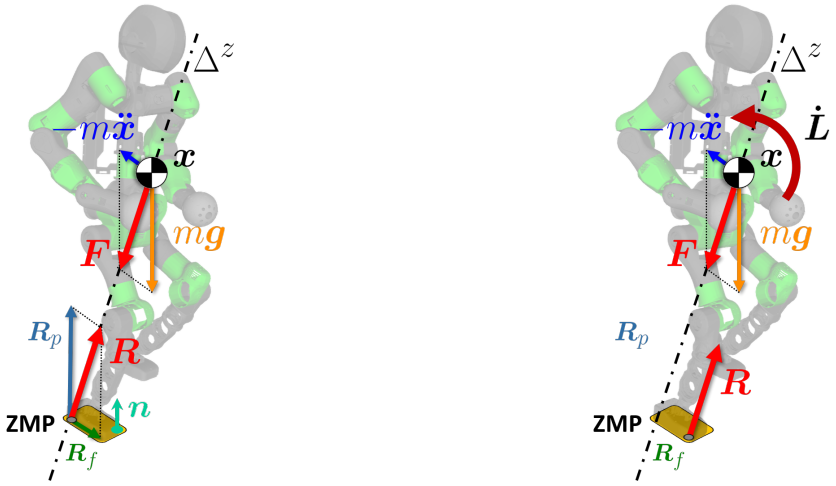
Pressure (CoP), shown in Figure 2.4. Therefore, it can be concluded that the CoM projection on the ground must lie inside the support polygon, and, if this is the case, it coincides with the CoP. By complying with this criterion, a simple *static walk* can be synthesized by slowly moving the CoM from the sole of one foot to another (Kato et al., 1974), to keep the CoM projection always inside the support polygon of the humanoid.

However, due to the heavy assumptions, static equilibrium is a notion far from general. As soon as the contact plane is not horizontal, or the contact points lie on non-coplanar surfaces, the decoupling in (2.35) between horizontal and vertical components is no longer available.

2.5.2 Zero Moment Point

A more flexible tool for analyzing the balancing of a legged robot was provided in Vukobratovic and Juricic (1969), which introduced the Zero Moment Point (ZMP). In short, the ZMP represents the point in the contact plane between the foot and the ground at which the reaction force does not produce any moment in the horizontal direction. Specifically, if

[2]The term "stability" applied to a legged robot coincides with the notion of "contact stability", a condition used to check the feasibility of the robot pose (or motion) in terms of equilibrium of the body on its limbs. In order to avoid confusion, here and throughout this work the term "stability" won't be used when referring to the balancing of a biped, as it may be mistaken with the well-defined notion of "Lyapunov stability", widely used in System Theory.



(a) (b)

Figure 2.5 Gravity and inertia forces and torques a the CoM and their projection of the ZMP. (a) If the angular momentum is conserved, the ZMP is the point at the intersection between the support polygon and the gravito-inertial axis. (b) When the angular momentum is not conserved, the axis shifts parallel to the one passing trough the CoM, so that the contact force at the ZMP generates the additional required momentum to counterbalance the rate of change of the system's angular momentum.

the robot is in contact with the environment with its feet only, and the reaction force in the system robot-environment consists of three components of the force $\boldsymbol{F} = \sum_k \boldsymbol{f}_k$ and three components of the moment $\boldsymbol{\tau} = \sum_k \boldsymbol{\tau}_k$. The forces acting in the horizontal plane (F_x, F_y) can be balanced by friction, while the vertical moment M_z represents the moment of the friction forces. If the assumption of no-sliding contacts is made, these components can be neglected. Term F^z is the ground reaction that balances vertical forces. The remaining components are the moments τ^x and τ^y, which correspond to the "tilting moments": unwanted moments in the horizontal plane of the foot would result in a rotation of the system about one edge of the foot and the robot would lose balance and tumble. Hence, if one point exists inside the convex region of the robot's support area at which the horizontal moments τ^x_{zmp} and τ^y_{zmp} are null, the stance is balanced. This is equivalent to write:

$$\tau_{zmp} \times \boldsymbol{n} = \boldsymbol{0} \tag{2.36}$$

where $\boldsymbol{n}$ is the normal to the support polygon. Formally, the ZMP can be obtained by following the same procedure as in Subsection 2.5.1 considering a non-static system. Thus, the linear and the angular momentum in the equation of the centroidal momentum (2.25) are

not discarded. In particular, the first equation can be written as:

$$\boldsymbol{p} = m\boldsymbol{g} - \sum_k \boldsymbol{f}_k \tag{2.37}$$

where $\boldsymbol{T}_C$ is the reaction force at the contact point C, and the remaining terms are the resultants of the gravity and the inertia forces at the CoM. The second equation can be expressed as:

$$\dot{\boldsymbol{L}} = \boldsymbol{x}_{com} \times m\boldsymbol{g} + \sum_k \boldsymbol{\tau}_k \tag{2.38}$$

where $\boldsymbol{L}$ is the angular momentum at the CoM and $\boldsymbol{\tau}$ are the torques at the contact points. These equations convey that the gravity and inertia forces must be counterbalanced by the reaction forces at the contacts for the robot to maintain equilibrium. Plugging (2.37) in (2.38) and solving for any contact point c the model becomes:

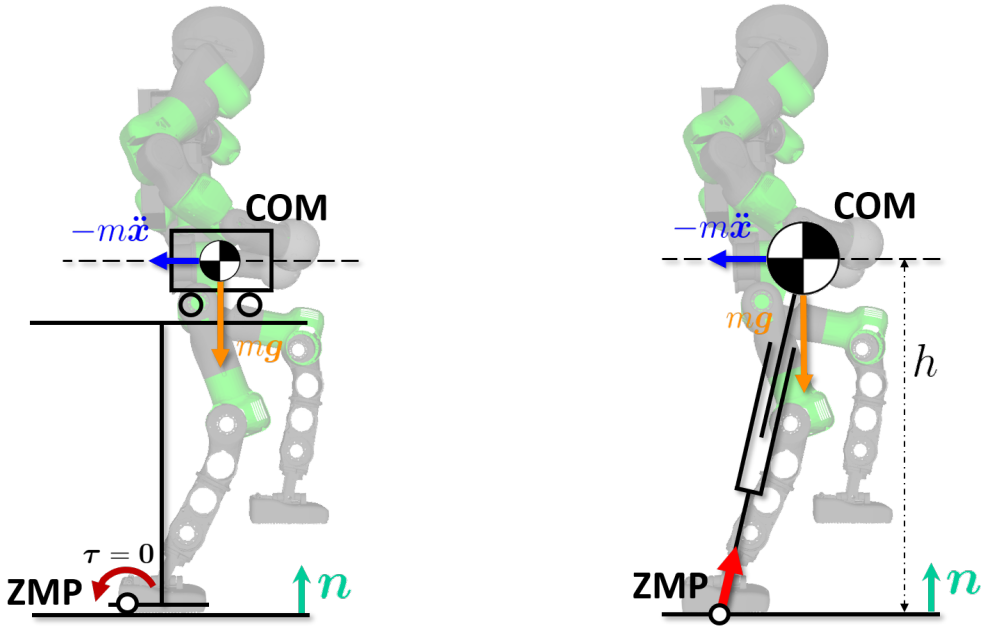
$$\boldsymbol{\tau}_c = \dot{\boldsymbol{L}} - \boldsymbol{x}_{com} \times m\boldsymbol{g} - \boldsymbol{x}_c \times (\dot{\boldsymbol{p}} - m\boldsymbol{g}) \tag{2.39}$$

Following the definition, the ZMP belongs both to the support region and to the axis Δ^z where the moment τ_{zmp} is always parallel to the normal vector $\boldsymbol{n}$. Therefore, the components $\tau^x_{zmp} = 0$ and $\tau^y_{zmp} = 0$, so:

$$\begin{aligned} x_{zmp} &= \frac{mgx_{com} + z_{com}\dot{p}_x - \dot{L}_y}{mg + \dot{p}_z} \\ y_{zmp} &= \frac{mgy_{com} + z_{com}\dot{p}_y + \dot{L}_x}{mg + \dot{p}_z} \end{aligned} \tag{2.40}$$

Note how the axis Δ^z pass trough the CoM only if $\dot{\boldsymbol{L}} = 0$, i.e. the angular momentum of the system is conserved (Figures 2.5a). On the contrary, if $\dot{\boldsymbol{L}} \neq 0$, the axis Δ^z shifts so that the reaction force at the contact point produces the counterbalancing angular momentum (Figure 2.5b).

This point, given some reasonable assumptions, can be exploited by controlling the position of the ZMP to achieve locomotion that preserves postural balance. The ZMP is broadly used in humanoids locomotion control for walking, using different strategies for gait balance (Kajita et al., 2003a), or running (Kajita et al., 2007), but the assumptions required by this criterion are limiting in more advanced situations. In fact, the ZMP approach can only be used for coplanar contacts between the mechanism and the environment. Moreover, the friction components are overlooked: at low friction, the robot could translate along the horizontal plane R_x, R_y or rotate around the normal direction to the ground M_z.



(a) Cart-table model

(b) Linear inverted pendulum model

Figure 2.6 Equivalent simplified models of a humanoid robot. (a) describes the ZMP trajectory w.r.t. to the CoM, while (b)

2.5.3 Cart-Table and Linear Inverted Pendulum Model

As formulated in Subsection 2.4.3, the model (2.25) is nonlinear. By recalling its definition in 2.5.2, the ZMP leads to two useful equations that can be exploited to simplify the system. As a matter of fact, the ZMP is the point on the contact plane where the moment of the resulting ground reaction has only a component in the vertical direction. Furthermore, by collecting all the contact forces of Equation (2.25) in one applied at the ZMP, the horizontal component of the resulting moment about the ZMP is equal to the total momenta generated by the body:

$$\begin{cases} \boldsymbol{n} \times \boldsymbol{\tau}_{zmp} = \mathbf{0} & (2.41) \\ \dot{\boldsymbol{L}} = (\boldsymbol{x}_{zmp} - \boldsymbol{x}_{com}) \times \boldsymbol{f} + \boldsymbol{\tau}_{zmp} & (2.42) \end{cases}$$

where $f = m(\ddot{\boldsymbol{x}} - \boldsymbol{g})$ is the resultant of the contact forces at the center of mass. The nonlinearity of the model is introduced by the angular momentum and the height variations of the CoM, a simplification of the problem can be easily obtained by making two assumptions:

(i) $\dot{\boldsymbol{h}}_L = \mathbf{0}$, the angular momentum at the CoM is zero;

(ii) $\boldsymbol{n}^\top \boldsymbol{x}_{com} = h$, the CoM height is constant.

Then, plugging (2.41) in (2.42) and imposing (i) yields[3]:

$$f_z\,(\boldsymbol{x}_{com} - \boldsymbol{x}_{zmp}) - h\boldsymbol{f} = \boldsymbol{0} \tag{2.43}$$

Considering the equation element-wise, the vertical component is just an identity, since $\ddot{z}_{com}$ has zero acceleration thanks to assumption (ii). The remaining equations represents the well-known equation of the *Linear Inverted Pendulum Model* (LIPM):

$$\ddot{\boldsymbol{x}}_{com} = \frac{g}{h}(\boldsymbol{x}_{com} - \boldsymbol{x}_{zmp})\Big|_{x,y} \tag{2.44}$$

This equation is linear and the ZMP is expressed only as a function of the CoM state and it only consider the planar linear momentum of the robot (Figure 2.6b). Often in literature the quantity $\omega^2 = g/h$ is called the *natural frequency* of the LIMP. Indeed, the exact result can be obtained from the general Equations (2.40), by noticing that, for a point-mass model, $\boldsymbol{p} = m\ddot{\boldsymbol{x}}_c$, and imposing assumptions (i) and (ii).

The LIPM representation is equivalent to the cart-table model shown in Figure 2.6a, which intuitively describes the problem of bipedal equilibrium. Indeed, when the cart moves close to the edge of the table the system starts tipping. However, given a proper acceleration, the table can keep its balance. What changes between the two models is the physical interpretation: the LIPM model is a proper differential equation describing the dynamics of the CoM given the ZMP an input 2.44, while the cart-table model swaps the causality, expressing the ZMP output as a function of the CoM, which corresponds to the ZMP equation 2.40 imposing the previous assumptions (i) and (ii):

$$\begin{aligned} x_{zmp} &= \frac{mgx_{com} + hm\ddot{x}_{com} - \cancelto{0}{\dot{L}_y}}{mg + \cancelto{0}{m\ddot{z}_{com}}} = x_{com} + \frac{h}{g}\ddot{x}_{com} \\ x_{zmp} &= \frac{mgy_{com} + hm\ddot{y}_{com} + \cancelto{0}{\dot{L}_x}}{mg + \cancelto{0}{m\ddot{z}_{com}}} = y_{com} + \frac{h}{g}\ddot{y}_{com} \end{aligned} \tag{2.45}$$

Note that a sudden acceleration of the CoM in one direction will produce a first motion of the ZMP in the opposite direction: systems that respond to a step input with an initial "undershoot" before converging to the steady-state are known as *non-minimum phase* systems. Plotting the evolution of this system over time reveals how this property may impact the control design. As shown in Figure 2.7, in a nominal walking where gait the ZMP is

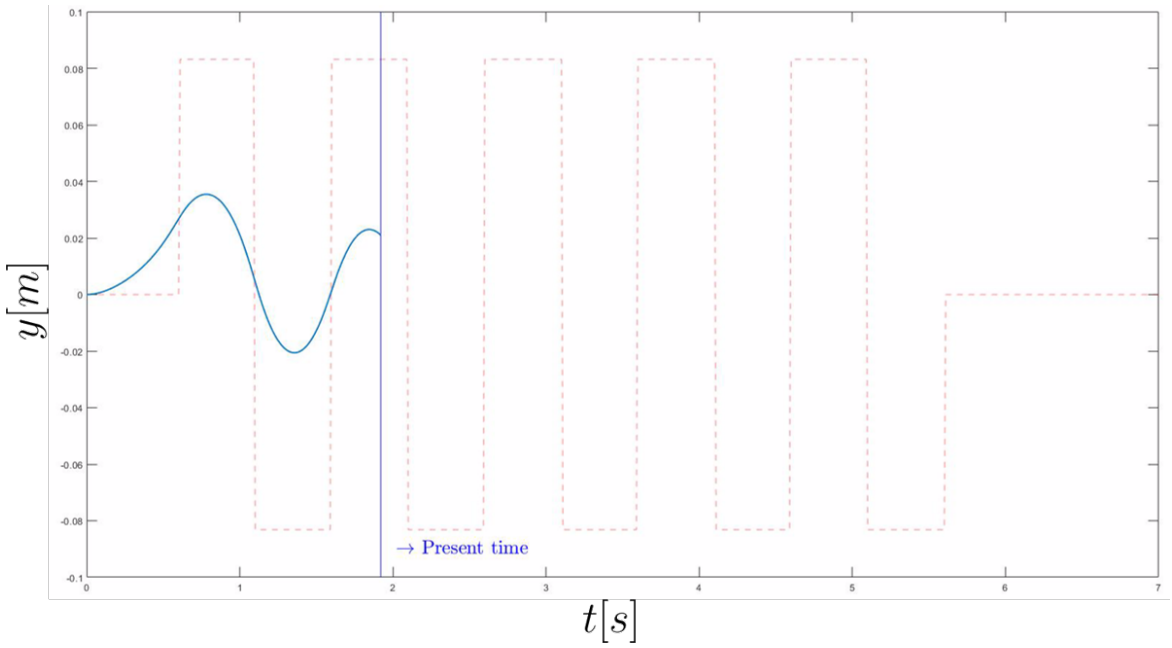


Figure 2.7 The reference trajectory of the ZMP (dashed line) shifts its position, alternating between the support area of the left and the right foot. For a balanced gait, the COM has to move *before* the robot changes the support leg.

commanded to stay in the center of the switching support polygon at all times, the ZMP moves from the foot in contact with the ground to the other. The resulting motion of the CoM occurs *before* the stepping motion, meaning that the relationship between the CoM and ZMP is anti-causal, since the output (CoM trajectory) depends on the future input (ZMP trajectory).

2.5.4 Friction cones

In previous sections emerged how contacts are a crucial component for humanoid motion and body equilibrium. In this respect, distinguishing between a fixed or a sliding contact is of equal importance: indeed, contact stability conditions can be obtained by modeling the contact of the robot end-effectors with the external surfaces. Once established the contact at the point C_i, the force f_c at the contact point is the following:

$$\boldsymbol{f}_c^n = (\boldsymbol{n}_c \cdot \boldsymbol{f}_c)\,\boldsymbol{n}_c \qquad \Longrightarrow \quad \text{normal component} \tag{2.46}$$

$$\boldsymbol{f}_c^t = \boldsymbol{f}_c - (\boldsymbol{n}_c \cdot \boldsymbol{f}_c)\,\boldsymbol{n}_c \qquad \Longrightarrow \quad \text{tangential component} \tag{2.47}$$

[3]The following equation is obtained using the vector triple product, i.e. $a \times b \times c = (a \cdot c) \times b - (a \cdot b) \times c$.

Intuitively, any mass in contact with a plane will remain fixed as long as the tangential force will not exceed a certain threshold. In particular, exploiting the definition of *Coulomb friction cone*, a contact is considered stable if and only if two conditions are preserved:

- $\boldsymbol{f}_c^n \cdot \boldsymbol{n}_c > 0$, it exists a component of the force in the normal direction;
- $\left\| \boldsymbol{f}_c^t \right\|_2 \leq \mu_c \left(\boldsymbol{f}_c \cdot \boldsymbol{n}_c \right)$, the tangential component lies inside the friction cone

where μ_c is the friction coefficient associated with the contact C_i and $\|\cdot\|_2$ models the isotropy of the friction, which is constant over the whole cone.

It is worth noting that a generalization of this concept can be used to prove body equilibrium of the whole robot. Recalling the centroidal momentum (2.25) and defining the *contact wrench cone* (CWC), which corresponds to the sum of all contact wrenches acting on the robot, contact stability is guaranteed only if the centroidal momentum $\boldsymbol{h} = \begin{bmatrix} \boldsymbol{p} & \boldsymbol{L} \end{bmatrix}$ lies inside the CWC. In other words, the aggregated wrench, consisting of the inertial component, the gravity component and the contacts with the external environment is to be balanced by the linear and the rate of change of angular momentum of the robot. This approach is equal to the ZMP when the contacts are co-planar, but it is not limited to foot-ground contacts and can be exploited for multiple contacts of the robot's body with the environment.

2.5.5 How stable is a humanoid robot?

Stability criteria such as static balancing, the ZMP or the CWC are useful to determine joint configuration and motions for contact stability of the humanoid. May be it on flat or uneven ground, for walking or in a multi-contact scenario, in the presence of external forces or without any force applied on the robot, these strategies are concerned with achieving body equilibrium. Then, the next natural question to raise is "how stable is a biped". In other words, a measure of how close a given humanoid robot is to fall should be provided.

Capture Point

A strong answer to the question is provided by Pratt et al. (2006), which proposes a novel tool for bipedal system equilibrium analysis. Informally, it defines the ability of a legged system to stop its motion without falling by eliminating the linear momentum. This definition is embodied by the *Capture Point* (CP), and it was extended building upon the notion of *viability kernel* (Aubin et al., 2011), which defines a state space for a system, divided into regions of capturable states and failed states. A state included in the capturable region, given the right input, will eventually come to a stop without falling, i.e. converge to a captured

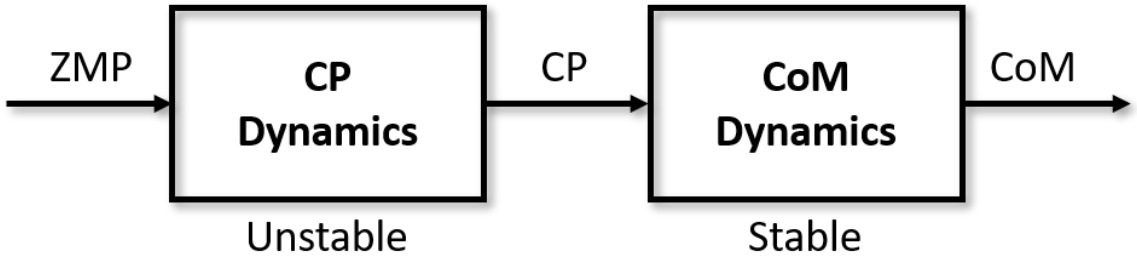


Figure 2.8 Scheme of the humanoid model dynamics. The CoM is attracted by the CP, which in turns diverges from the ZMP. Controlling the robot to prevent the CP to diverge will stabilize the system, as the passive dynamics of the CoM will naturally follow the CP.

region (or in very specific cases, end up in an infinite limit cycle that prevents the motion from coming to a stop to keep equilibrium). Different inputs can push the state outside of the capturable states, ending up in a failed gait which will result in the robot tumbling. The CP is then is the point on the ground where the legged system should place its foot to come to a stop. When the CP lies outside the reach of the robot, more steps are required, i.e. the system is in a N-step capturable region and, if the right steps are taken, it can recover balance. Formally, the CP can be derived from the equation of motion of the system in (2.44), as it is deeply connected with the ZMP. The explicit solution of the second-order differential equation corresponds to the sum of three components:

(i) $\boldsymbol{x}_{zmp}$, a stationary term corresponding to the position of the ZMP;

(ii) $\lim_{x\to+\infty} \boldsymbol{A}_1 \frac{e^{\omega t}}{2} = 0$, the stable component of CoM motion;

(iii) $\lim_{x\to+\infty} \boldsymbol{A}_2 \frac{e^{-\omega t}}{2} = \infty$, the unstable component of CoM motion.

where, given the initial condition $[\boldsymbol{x}_{com}(0), \dot{\boldsymbol{x}}_{com}(0)]$:

$$\boldsymbol{A}_1 = \boldsymbol{x}_{com}(0) + \frac{\dot{\boldsymbol{x}}_{com}(0)}{\omega} - \boldsymbol{x}_{zmp} \qquad \boldsymbol{A}_2 = \boldsymbol{x}_{com}(0) - \frac{\dot{\boldsymbol{x}}_{com}(0)}{\omega} - \boldsymbol{x}_{zmp} \tag{2.48}$$

Thus, if the effect of the divergent component is canceled, the trajectory of the CoM will naturally converge, since term (ii) is stable. This brings to the definition of the CP in the divergent component (iii):

$$\boldsymbol{x}_{cp} = \boldsymbol{x}_{com} + \frac{\dot{\boldsymbol{x}}_{com}}{\omega} \tag{2.49}$$

The only bound CoM trajectory corresponds to the one where the ZMP is at the CP, i.e. the point where the ZMP should move instantaneously to stop completely the robot motion.

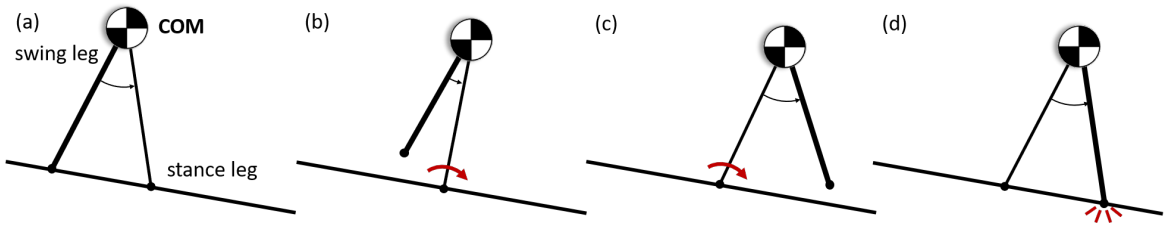


Figure 2.9 Simplified model of a passive walker performing one single walking step, which corresponds to a full cycle in the phase space. The robot configuration at the impact (d) is the same as the initial configuration of the step (a), which defines a periodic motion, i.e. *limit cycle walk*.

This is also known as the *boundedness condition* formulated in Lanari et al. (2014), which states that the CoM does not diverges only if a suitable initial condition of the ZMP is provided $\boldsymbol{x}_{zmp} = \boldsymbol{x}_{cp}(0)$. The CP is not directly related to the forces exerted on the system, but there is a tight relationship between the dynamics of the CP and those of the CoM. If the CP is considered not as a point but as a system, its dynamic can be derived decomposing the second-order dynamics of the CoM into two first-order equations. By differentiating Equation (2.49) and plugging in (2.44):

$$\begin{cases} \dot{\boldsymbol{x}}_{com} = \omega(-\boldsymbol{x}_{com} + \boldsymbol{x}_{cp}) & (2.50) \\ \dot{\boldsymbol{x}}_{cp} = \omega(\boldsymbol{x}_{cp} - \boldsymbol{x}_{zmp}) & (2.51) \end{cases}$$

This formulation decouples the exponentially stable dynamics of the CoM, which is constantly drown to the CP, and the unstable dynamics of the CP, which tends to diverge from the ZMP, as shown in Figure 2.8.

Limit Cycle Walking

In previous sections it was presented and discussed the equilibrium of a humanoid robot. This concept can be defined in different ways and under different assumptions, hence, the generic goal of "not falling" can be attained following different criteria. However, this notion is not very useful for gait synthesis, as it can be formally defined in many different ways.

Two opposed strategies lie at the extrema of the same concept: the first is the most restrictive criterion, as it only considers motions that are stable along the whole trajectory. It is known in literature as *sustained local stability*, since the equilibrium of the robot is continuously enforced. This notion usually coincides with the ZMP criterion, and it has been proven to be extremely useful in gait synthesis, but is often “clumsy” and under-achieving

compared to the human walk, which, on the contrary, exploits the natural falling dynamics of the body to move more efficiently. The diametrically opposite strategy propose a very general indicator, known as the "viability kernel" 2.5.5. This set of states includes any possible motions of the robot which results in self-balancing, even periodic motions or states perturbed by external disturbances. However, applying this criterion means checking every state in the state space of the robot for determining if it is viable or not: due to the high non-linearity and the complexity of the full model, exploiting effectively the viability kernel is computationally impossible.

One branch of humanoid locomotion research distance itself from the sustained local stability, focusing on the motions of the viability kernel that results in periodic gaits. Known as "*Limit Cycle Walking*", this criterion releases the constraints of the ZMP by considering periodic gaits, and consequently balanced in the long term, which are not locally stable. In other words, gaits that entail portions of the walking where the robot may be considered "falling".

This criterion can be formalized by defining one cycle as a mapping from the robot state p_n to the same state p_{n+1}, where a state s represents a given point in the motion of the robot, such as the moment of taking off of the foot in a nominal walking. This mapping is usually realized by using the Poincaremap, which is defined by integrating the equations of motion of the robot over one full cycle. Without requiring any other assumption on self-balance, if this point exists:

$$p_{n+1} = \boldsymbol{S}(p_n) \tag{2.52}$$

then it exists a limit cycle which can be physically interpreted as a feasible walking gait, as depicted in Figure 2.4. In particular, Poincaremaps are used in Hobbelen and Wisse (2007) to identify regions of stability and measure the stability of limit cycles representing gaits.

Part I

Hardware and Software Level

Chapter 3

Robotic Platform: the Humanoid COMAN+

3.1 Mechanical overview

The robot used for simulation and real-hardware experiments is COMAN+, a humanoid robot developed at Istituto Italiano di Tecnologia by the Humanoids & Human Centered Mechatronics Lab (HHCM).

COMAN+ is the third generation of humanoid robots built at the HHCM, its predecessors being COMAN (Tsagarakis et al., 2011) and WALK-MAN (Tsagarakis et al., 2017). The current platform introduced several improvements to the robot performance while inheriting many traits of the previous models. COMAN+ has a very similar body structure to the average adult human being: its total weight is 70 kg, its height is 1.70 m, the width between the shoulder is 0.6 m. The structure of the robot is depicted in Figure 3.1. The only body part which deviates from the anthropomorphic ratio is the arm: while the human arm's span, i.e. the distance between the left and the right hand when stretched out horizontally, is approximately one-to-one compared to its height, COMAN+ ratio is roughly 1.2. The choice of increasing the arm length dates back to the humanoid WALK-MAN, and finds its main reason in a configuration ideal for disaster scenarios: supported by strong motors that can cope with their extended length, longer arm effectively enlarge the manipulation workspace of the robot, allowing to reach less accessible areas or surfaces. Furthermore, the proximity of the hands to the ground facilitate their positioning for a quick response against instabilities. Disaster scenarios are usually cluttered areas composed of uneven terrains and rough surfaces. With the aim to operate in such an environment, COMAN+ was designed following three leading criteria to improve its physical interaction performance:

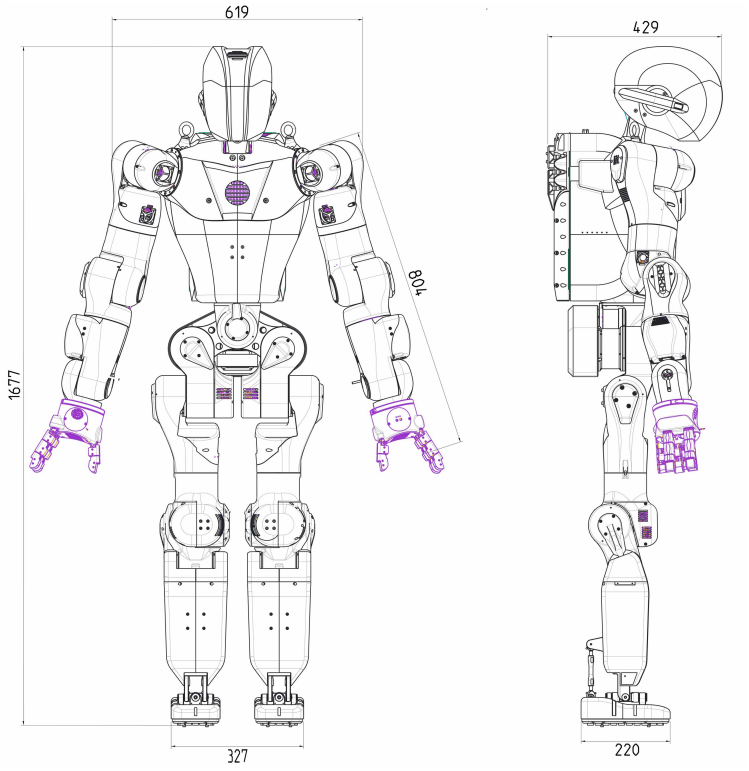


Figure 3.1 Outline of the humanoid robot COMAN+.

- *passive compliance*: Series elastic actuation
- *physical sturdiness*: custom-made actuation system
- *high strength capabilities*: strong torque of actuators /high-performance modules heavy manipulation
- *human-like motions*: large joint range of motion

These attributes are provided by the custom-made actuation system, the limbs design and the sturdy yet light skeleton.

3.1.1 Actuation

The actuators at each joint are fully developed at IIT: they consist of brushless DC motors connected to a harmonic drive gearbox and a fixed elastic element positioned between the gearbox output and the load, following the Series Elastic Actuator (SEA) scheme. The design of the motors has been optimized in terms of dimension and weight, while conforming its nominal and peak torque of the actuators to the requirements of a full-sized humanoid.

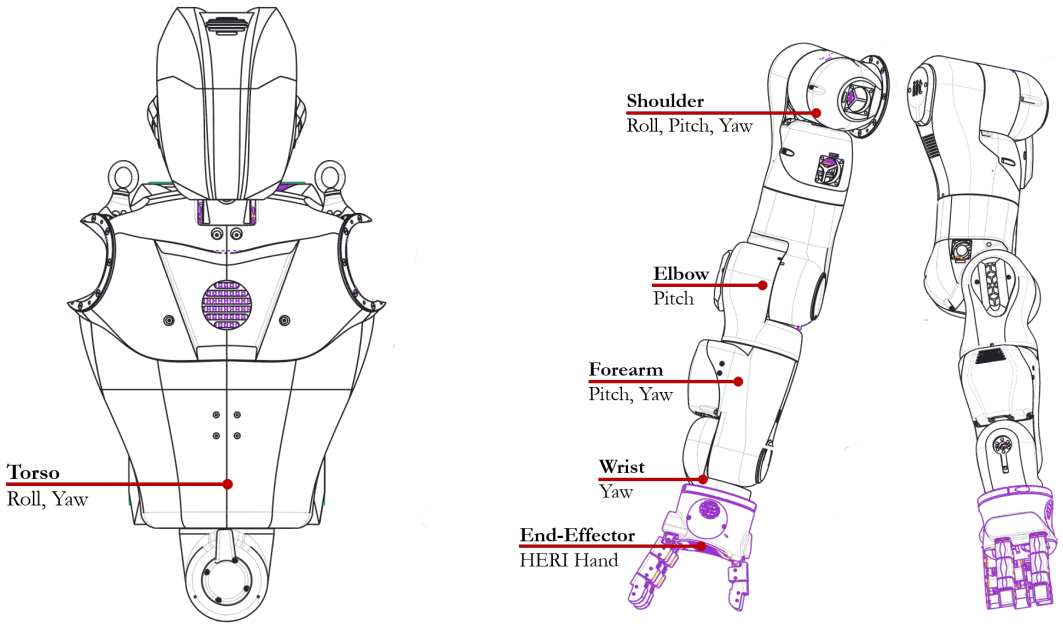
Figure 3.2 The family of torque controlled actuators used in HHCM robots, covering a wide range of torques, speed and weights.

The resulting semi-flexible body structure cues to the intrinsic elasticity of human muscles. These actuators provide additional physical protection of the transmission against high bandwith impacts. This mechanical compliance is also useful for adaptation to the environment and moderate disturbance rejection. The deflection of the torsional bar is measured by strain gauges, allowing for improved torque sensing and enables high-fidelity torque control. Similarly to its previous version WALK-MAN, this platform is composed of 28 degrees of freedom: 7 joints for each arm, 6 for each leg and 2 for the torso. The motors at each joint come in five different sizes and transmission ratios, encompassing the requirements for weight, speed and torque output.

Each actuator, besides the intrinsic torque-sensing capabilities, incorporates one encoder to measure the absolute joint position and a temperature sensor to monitor its state. These high-performance modules reliably deliver strong torques while being compliant and robust: characteristics ideal not only for physical interaction tasks but also for heavy manipulation. The robot has a slot for the power pack for autonomous walking.

3.1.2 Upper body

The torso of COMAN+ can rotate along the yaw and the roll axis, enabling motion in the transverse and frontal plane. Indeed, the choice of removing the DoF in the sagittal direction diminishes the range of motion of the robot, but the lack of this DoF can be handled by the robot's pelvis, composed of the first pitch joint at each leg. The arm's structure follows the nominal anthropomorphic design, providing kinematic redundancy w.r.t. any manipulation and interaction task that require setting position and orientation of the end-effector. Three actuators compose the spherical joint at the shoulder, one pitch joint is located at the elbow and one yaw enables forearm rotation. The wrist is made of two DoF, one pitch and one roll. A depiction of the kinematic structure of the robot is shown in Figure 3.3. The robot



(a) (b)

Figure 3.3 Kinematic structure of COMAN+ torso and arm.

can be equipped with different end-effectors acting as hands. Some interchangeable devices have been developed depending on the nature of the task that the robot is carrying out: the anthropomorphic soft-hand (Catalano et al., 2014) is useful for delicate interaction, while two simpler end-effectors can be mounted for tasks that do not require complex manipulation, one consisting of ball-hands and the other of metal plates. Both tools are rubber-coated to reduce shocks and increase friction, essential features for physical interactions with the environment or heavy manipulation. In particular, each arm can manipulate over 10 kg over the whole workspace, demonstrating high power density while keeping excellent dexterity.

3.1.3 Lower body

The lower body is designed to resemble the human structure and mimic its relative dynamic performance. For these reasons, agility and robustness are the main goals to pursue in the development of the legs. In particular, the peak velocity and acceleration are to be maximized, while limiting the weight and the inertia of each leg. Furthermore, their architecture must withstand impacts, minimizing mechanical failures or malfunctions. Each leg presents a spherical joint at the hip, one DoF at the knee and two DoF at the ankles, as shown in Figure 3.4. A particular four-bar mechanism controls the pitch and the roll at the ankle level

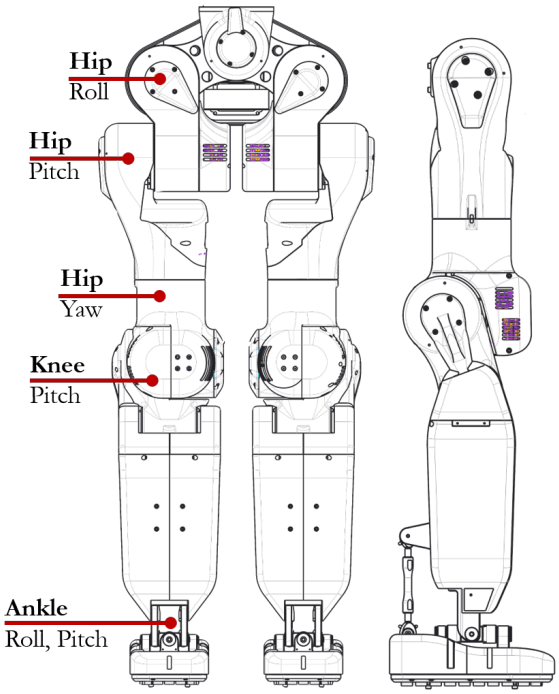


Figure 3.4 Kinematic structure of COMAN+ legs.

of the robot: the actuators are on the shin, reducing the inertia at the feet. Furthermore, the leg mass is concentrated at the hip to further reduce the inertia of the limbs. This allows faster swing motion and reduces coupling between the lower-body and the upper-body movement. Since the foot is subject to frequent and consecutive impacts, the sole is covered with a rubber layer for shock absorption and to increase its grip on the ground. To monitor the occurring impacts and their magnitude, each foot is equipped with a six-axis force-torque sensor. At the pelvis level has been installed an Inertial Measurement Unit (IMU).

3.1.4 Parallel kinematic ankles

To minimize weight and inertia at the extremity of the legs, their mass should be concentrated as close as possible to the waist, where the CoM of the robot is located. Having high inertia at the extremity of the limbs, especially on the feet, is detrimental to the robot's motion capabilities. Having high masses close to an end-effector prone to frequent impacts increase the magnitude of the shocks that the robot receives, fostering unwanted rebound that could destabilize the robot and raising the risk of system failures, particularly undesired for robots dealing with disaster scenarios where safety is essential. Furthermore, a greater lever arm

slows down the system response, limiting dynamic movements and reactive motions. Finally, high inertia translates to higher energy consumption and higher torques, which in turn entails bigger and heavier motors.

For these reasons, the leg was specifically designed to reduce the elements at the ankle level: the actuators of the pitch and roll joints are positioned above, along the shin, and connected to the ankle with a *parallel mechanism*. The motors actuation is transferred to the parallel mechanism by two rubber bands.

The classic parallel mechanism is composed of one or more kinematic loops connecting a moving component to a fixed base. These parallel chains are suitable for the typical configuration of the ankle, which is composed of a roll and a pitch DoF. Compared to a serial motor arrangement, the torque profile is preferable. Since the ankle pitch requires higher torques than the roll due to the structure of an anthropomorphic body, the size of the relative actuator should be bigger. The parallel mechanism distributes the load evenly, lifting the requirements on the size and granting the possibility to use identical motors. Furthermore, thanks to the transmission bars, the weight can be shifted upwards along the tibia, moving the actuation system away from the ankles, and therefore increasing the dynamics capabilities of the swinging leg. Besides, given the same motor size, this result allows for higher torque outputs, that further improve the dexterity of the robot.

Indeed, the downside of this choice is a narrower range of motion of the mechanism: however, considering the requirements of the ankle joints, power distribution, accuracy and performance are to be preferred. Moreover, it was demonstrated how this configuration allows for a constant torque transmission within the range of motion of the ankles.

The benefits of this approach come at the expense of the complexity of the system: as a matter of fact, the kinematic analysis of the mechanism and its control are not straightforward due to the coupling between the two DoF and the distancing between the motor and the joint. The displacement of one motor does not directly correspond to the motion of one degree of freedom: on the contrary, to generate a pure pitch, both motors need to operate in the same direction (*common mode*), while a pure roll requires one motor to rotate in the opposite direction of the other (*differential mode*). To control the full range of motion of the ankle, the synergy of the motors must be mapped. This amounts to extract a Jacobian matrix that maps the motor space to the joint space, as the joint and motor angles are related through a forward kinematic mapping.

3.2 Software

COMAN+ is powered by XBotCore (Muratore et al., 2017), a robotic middle-ware that connects the lower-level control to the custom control layer written by a developer. It has been conceived to fulfill the requirements of hyper-redundant robots executing performance-critical applications: the XBotCore architecture ensures hard real-time (RT) control performances, granting a high control frequency of 1 kHz even for complex hyper-redundant systems, reducing latency and jitter. This aspect is particularly important for demanding physical tasks. The structure of XBotCore was kept as intuitive as possible, bearing transparency and modularity as critical prerogatives of the architecture. These features are especially useful for diagnostic purposes and code development in inherently complicated systems, where any overhead complexity should be minimized. This software is cross-compatible with any robot given minimum code adaptation. Its flexible nature allows for plug-ins to be developed by third parties and integrated into the system as specific modules. With this respect, XBotCore exposes an independent abstraction layer that can be modified according to the type of robot hardware to be integrated. By providing a specific implementation to some abstract method, it is possible to establish a communication between high-level software and the robot, regardless of its specific structure. Other than real robotic platforms, XBotCore integrates an implementation for the Gazebo simulator environment, useful for analyzing the robot behaviour before deploying any control strategy on the hardware. XBotCore incorporates the possibility to leverage on ROS-like concepts such as topics and services both in non-RT and RT environments. Exchanging data within the RT domain is less straightforward, but it allows taking full advantage of the low latency of the system. The infrastructure grants a flexible Input/Output (I/O) interface which allows requesting data and sending commands to programs running in RT loops using the well-established paradigm of callbacks from publishers, subscribers or service servers introduced by ROS.

3.3 Model description

A popular middleware used in the HHCM Lab is Robot Operating System (ROS), which provides a convenient and flexible environment to develop robotic applications. Working with a specific robot in ROS requires the specification of its model in standard formats: the Unified Robot Description Format (URDF) and the Semantic Robot Description Format (SRDF). A URDF defines the structure of a robot, specifying the kinematic and dynamic properties using "*Elements*" and "*Attributes*". The relevant parameters are organized in a

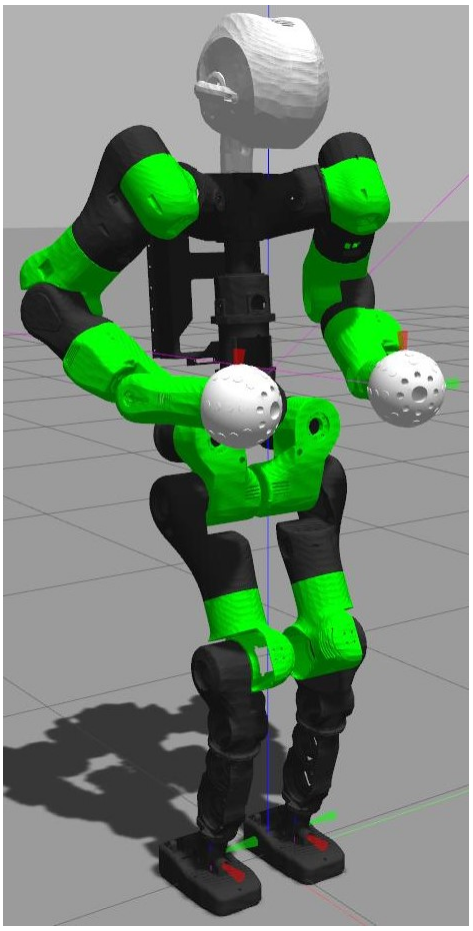
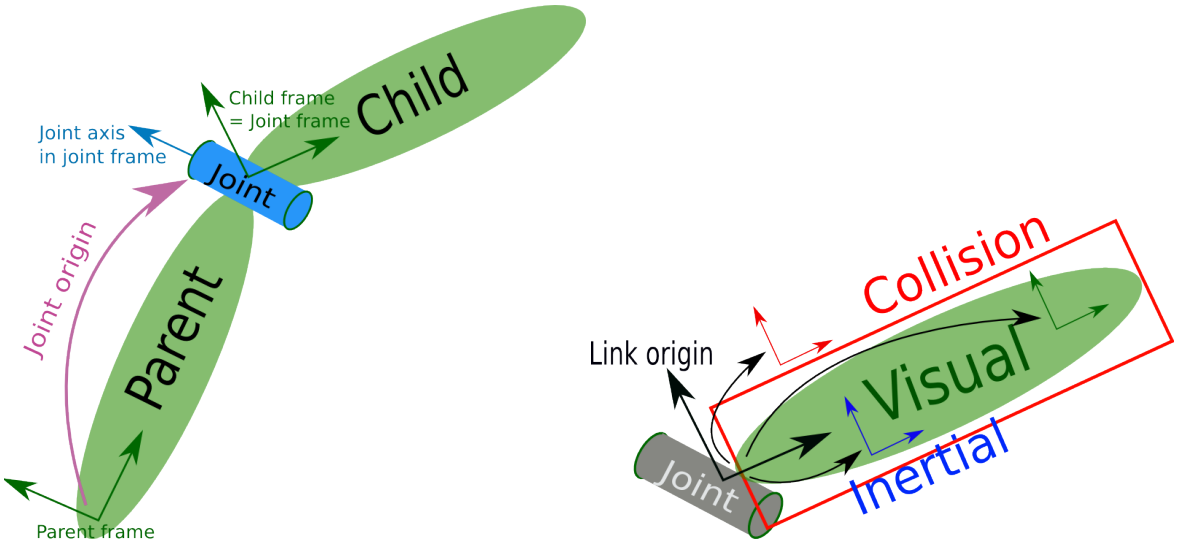


Figure 3.5 Model of COMAN+ simulated in the environment GAZEBO.

hierarchy to fully describe the hardware and the connections between the different parts. The URDF has a nested structure, where each *Element* is characterized by *Attributes* and can include other sub-elements. Some examples of the members of a URDF are:

- *Elements*: `<Robot>`, `<Joint>`, `<Link>`, `<Sensor>`, `<Transmission>` ...
- *Attributes*: `name`, `type`, `rpy`, `xyz`, `damping`, `friction`, `effort` ...

The SRDF complements the URDF and provides additional information for the robot model, such as the definition of virtual elements, collisions, groups of elements, etc. Indeed, for an articulated system, the most important elements are the `<Link>` and the `<Joint>`. The `<Link>` is described by the elements `<Collision>`, `<Visual>`, `<Inertial>`, which describes its kinematics and dynamics attributes and by the mandatory attribute `Origin`, positioning it in space w.r.t. the preceding joint (see Figure 3.6b). The `<Joint>`, in turns, is defined by the `<Parent>` and the `<Child>` links connected by the joint, the `Origin` w.r.t.



(a) <Joint> element. The joint connects the predecessor (parent) link and the successor (child) link. The origin is defined w.r.t. the parent frame.

(b) <Link> element. It is characterized by a <Collision>, <Visual> and <Inertial> element. The origin is coincides with the joint origin

Figure 3.6 Scheme of the joint and the link element description in the URDF format. These images are taken from http://wiki.ros.org/urdf/XML

the parent frame and the <Axis> specifying the direction of the DoF. The axis is specified in the joint frame of reference, while the child link frame coincides with the joint frame, as depicted in Figure 3.6a. A useful feature introduced by ROS is the language XACRO (XML Macro), which allows assembling the URDF as a collection of macros that expand to verbose XML expressions. XACRO makes the URDF more readable and flexible, as it uses variables that can be assigned to the properties, math expressions, simple conditional blocks, default parameters, and the possibility to include other XACRO files. When fed to ROS, the XML description is read by a dedicated parser, which transforms the list of elements and attributes in the corresponding kinematics chains of the robot. Once built, the robot is part of the ROS pipeline: its parsed structure can be retrieved as a tree graph, it can be visualized in 3D viewers such as RVIZ (Figure 3.7), simulated in GAZEBO (Figure 3.5) and, in general, it can be used in custom applications by the developer.

3.4 Contribution

In a mechatronics environment such as the HHCM lab, where different robots are developed and controlled, it is often required to access their model. In fact, any hardware alteration should be followed by the corresponding update of the URDF model. For instance, a

modification in a component (e.g. weight, shape, etc.) corresponds to an adjustment of the relative parameters. Similarly, substituting a joint in the robotic hardware requires the same substitution in the model. Furthermore, as the robot structure grows in complexity, the URDF file tends to become very large and, consequently, difficult to read and maintain. While a hierarchy of XACRO simplifies the process, sometimes macros are not powerful enough to handle a complex robotic platform such as a humanoid.

The need for a simple yet powerful tool to overcome these issues led to the development of *RoboGen*, a dedicated library for URDF and SRDF generation, designed with the following features in mind:

- *powerful and intuitive*. Exploits the potential of a programming language instead of XML format or XACRO hierarchies;
- *one-to-one mapping* from XML format to Python language: elements and attributes in the XML format are Python classes and classes' attributes that can be added, deleted and modified at any time;
- *fully customizable* elements and re-usable blocks for sequential model construction
- *off-the-shelf elements generation* using the *urdf-factory*, so that template *Elements* can be created without specifying every detail
- *deployment* of the robot model to its XML equivalent at any time, both as a file or as a string for direct integration with the code
- XML *robot model extraction* and conversion into Python format with *urdf-parser*: the URDF can be imported in XML format for modifications.

RoboGen allows generating programmatically any URDF and SRDF file with a simple Python interface, instead of manually writing the XML or XACRO document. It is based on the library *etree* from the *lxml* package [1], an API for parsing and creating XML data. As discussed in last section 3.3, a URDF element is characterized by some properties that can be required or allowed:

- *attributes*
- *sub-elements*

[1] https://lxml.de/index.html

The sub-elements are, in turns, elements which can have more nested sub-elements. Every element in the XML format has a corresponding Python class of the same element type. For example, a `<Link>` has a corresponding class `Link()`, which is used to specify a desired *instance* of the element link. All classes are derived from the parent class `Element()`. Each element can be built from the constructor or populated sequentially, depending on the need. If desired, elements can be modified by adding or removing sub-elements and attributes. To avoid the generation of faulty URDF documents, *RoboGen* throws an error if a *required* sub-element (or attribute) is omitted when instantiating a parent element. The same happens if the user attempts to remove a *required* element or add a *non-allowed* element (or attribute).

Duplicating an URDF is very common when dealing with robots used for different tasks in different scenarios: instead of changing the same URDF, a set of URDF files covering the different robot version is used. Furthermore, the modular nature of the robots built at the HHCM Lab asks for a method to define a group of element definitions that can be re-used as components for another robot. For these reasons, a `Collection()` element is available that allows collecting multiple elements as one single component. At any time the elements can be converted to the XML format, which can be easily dumped to screen for debugging. However, the final XML object is often awkward to read. Therefore, any element in *RoboGen* comes with two useful methods to inspect the attributes and the elements. This resource is extremely useful when building a robot, since it shows all the types of sub-elements of an element, listing the name (if a name is given) of each particular instance. Additionally, it declares the position of each item, so that they can be easily accessed. The available methods in *RoboGen* are listed below:

- `.add_elements() / .add_attributes()`: add an element/attribute to an existing element, if allowed;
- `.remove_elements() / .remove_attributes()`: remove an element/attribute from an existing element, if allowed;
- `.get_elements() / .get_attributes()`: get the pointer to an element or a string of the attribute from an existing element;
- `.inspect_elements() / .inspect_attributes()`: returns a simple list with all the elements/attributes of an existing element. In the case of elements, it also returns the class of each element listed;
- `.sort_elements()`: reorder the sub-elements. By defaults it sort them alphabetically.

- `.get_xml()`: returns a pointer to a *xml* object that can be dumped on screen or printed to file.

Some use cases for these methods and their relative output are illustrated in Listing 3.1. While this library greatly simplifies writing the URDF, the nested nature of the structure is inherently burdensome. A hyper-redundant robot has a plethora of elements, each of which is composed of many sub-elements and attributes, taking a long time to define. To further assist the user in the process, *RoboGen* provides the *urdf-Factory*, a tool to create template elements without specifying each every detail. This is particularly advantageous when generating prototypes that can be detailed afterward, adding or modifying some properties. The *urdf-Factory* contains many default elements, such as:

- `.generate_element(Joint)`: if not specified otherwise, the default joint is revolute;
- `.generate_element(Link)`: the most common `<Link>` is just a blue cylinder;
- `.generate_element(Module)`: the union of a `<Joint>` and a `<Link>`;
- `.generate_element(Chain)`: a collection of *Modules* forming a chain with a base link. The order of the DoF type (i.e. roll, pitch, yaw) can be easily prescribed. If specified, an end-effector is added at the end of the chain, which includes a virtual frame at the tip, usually useful when controlling it.

Indeed, the constructor of each unit accepts any allowed parameters for customization. The following line of code returns a complete *module* by only specifying some relevant attributes:

```
.generate_element("Module", name="a_module", mass=2, revolute_type="
    pitch", parent="parent_link")
```

Finally, the *urdf-Parser* provided by the library allows parsing a URDF and transform it in a *RoboGen* class: when it is deemed useful to convert a robot model already available in an XML file to a more flexible format, this tool rapidly analyzes the URDF and gathers all the data and store it in a Python object that can be accessed at will.

3.5 Discussion

The library *RoboGen* proved to be very useful for the generation of both prototypes and real robots, such as COMAN+ (see Figure 3.7) and CENTAURO (see Figure 3.8). It considerably reduces the time to write a URDF description, and, consequently, the errors that go unnoticed in the XML. Using a programming language instead of XML or XACRO offers many benefits:

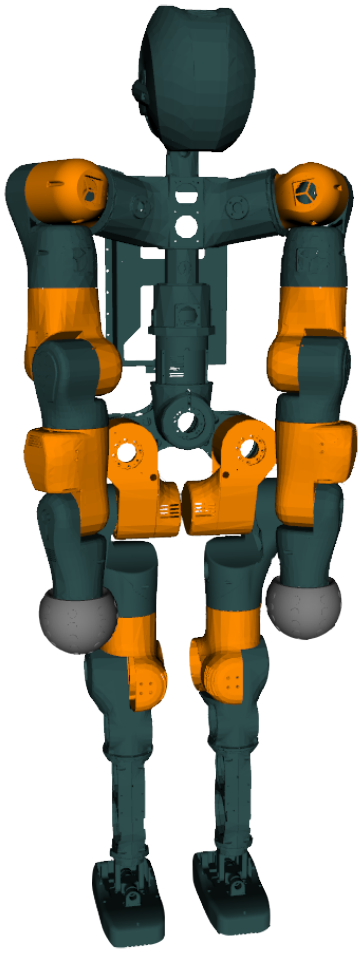


Figure 3.7 3D visualization in RVIZ of COMAN+ generated with *RoboGen*.

the debugging phase is significantly easier and the unit tests are more straightforward to design. Moreover, the structure of the Python script is completely up to the user. Being Python scripts, each file describing a collection of elements can be wrapped in a class that can be easily reused in other projects as off-the-shelf standard components. More importantly, the full potential of a programming language can be exploited. Therefore, besides benefiting from the basic features of data processing, the model generation can be integrated with other programs, such as parameters optimization, structural analysis and or dynamic simulations. The library has become stable and reliable, and it is now used for storing and generating the models at HHCM Lab. Future works will expand the pool of methods that *RoboGen* makes available, adding the possibility to retrieve high-level information about the robot, such as its total mass, its work-space and the available torques it can exert, based on the kinematics, the joint limits and the motor's specification. Furthermore, the possibility to

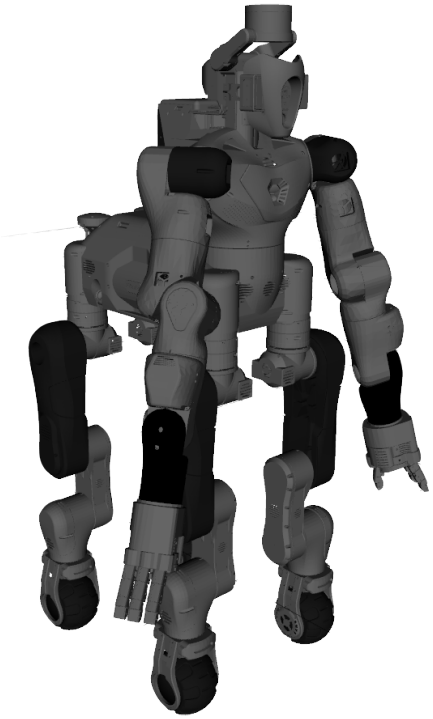


Figure 3.8 3D visualization in RVIZ of CENTAURO generated with *RoboGen*.

select the desired model among templates "skeletons" will be added, e.g. biped, tripod, quadruped, insect-like or wheeled robot, that can be subsequently personalized. While this is mostly aimed at prototyping, another future direction will target the integration of *RoboGen* with the design software used to model the real robot's parts, to automatize the transition between 3D modeling and URDF spawn.

Listing 3.1 Use cases of the library *RoboGen*.

```
# sub-module "urdf_elements" gives access to all the elements
import urdf_elements from robogen as urdf
# generate an instance of <Link>
link_elem = urdf.Link(name='a_link')
# generate <Inertial> element (sub-elements: <Mass> and <Inertia>)
inertial_elem = urdf.Inertial(elements=[mass_elem, inertia_elem])
# add "inertial_elem" to link_elem and dump it to screen
link_elem.add_element(inertial_elem)
etree.dump(link_elem.get_xml())
```

```
<link name="link_elem">
  <inertial>
    <mass value="2"/>
    <inertia ixz="0" ixx="1" ixy="0" izz="1" iyy="1" iyz="0"/>
  </inertial>
</link>
```

```
# inspect "first_link" to show its attributes
first_link.inspect_attributes()
```

```
name: first_link
```

```
# inspect "first_link" to show its elements
first_link.inspect_elements()
```

```
Collision:
   2) one_collision_element
Inertial:
   0): --
Visual:
   1) one_visual_element
```

```
# different ways to get the same element
no_named_inertial = link_elem.get_elements(0)
same_element_as_above = link_elem[0]
print no_named_inertial
print same_element_as_above
```

```
<robogen.urdf_elements.Inertial object at 0x7f17440bea50>
<robogen.urdf_elements.Inertial object at 0x7f17440bea50>
```

Part II

Low-level Control

Chapter 4

Fail-safe Impedance Control

The great progress robotics underwent in recent years promoted the transition from specialized robots specifically built for industrial tasks in controlled environments to general-purpose platforms that operate in unstructured locations under variable conditions, deals with obstacles unknown a-priori, and share their work-space with people. Especially for humanoids, among many desired features two aspects become critical, stemming from the need to interact frequently with the environment and with humans: compliance and reliability. Interacting effectively with the environment is very relevant to humanoids and in general to legged locomotion where the ability to conform and adapt to ground disturbances and impacts is a fundamental prerequisite for movements that preserve postural equilibrium. Hence, the first characteristic a humanoid robot should possess is the ability to comply with its surrounding, essential to achieve dexterous and agile motions displayed the same capabilities of their biological counterpart, as well as reduce the possibility of life-threatening impacts with human being. The higher autonomy expected from these machines goes hand in hand with the safety they must guarantee, as critical failures in uncontrolled settings may damage properties or worse, harm humans. Increasing the versatility of a robot implies rising its complexity while maximizing its reliability, both concerning hardware and software. This is particularly true for humanoids designed for disaster scenarios, carrying out complex tasks and working side to side with human workers. A hyper-redundant mechanism is composed of many actuators driven by as many processors unit, communicating together through several interfaces. A refined sensory system monitors the performance of the robot and is essential for accurate interaction with the environment. A dedicated software is required for coordinate the motion of the whole system, receiving data from the sensors, processing them and sending back to the actuators suitable signals while complying with time constraints and dealing with information loss. All these factors contribute to the intricacy of a system which should be as

autonomous as possible: therefore, tightening the safety measures and reducing the failure modes is a crucial point for the successful deployment of the robot. Besides developing a robust and reliable mechanical structure, great effort was put into the design of the system's software, meeting the requirements discussed above.

The SEA actuators of COMAN+ guarantees a certain amount of mechanical compliance to the robot's structure. Nevertheless, robust locomotion and complex interactions require a layered framework that spans from lower-level active compliance, such as admittance controller (Li et al., 2012; Zhou et al., 2014), joint impedance control (Mingo Hoffman et al., 2013), Virtual Model Control (Pratt et al., 2001) to high level reactive behaviours, e.g. footstep planning (Chestnutt et al., 2005; Herdt et al., 2010a) or balance recovery (Stephens, 2007). State-of-the-art robots are endowed with compliant architecture (passive or active), but not many can withstand potential system failures that could compromise the hardware integrity or the safety of humans interacting with the robot. The next section focuses on the first aspect, presenting the required background, while the following sections are devoted to an in-depth presentation of the contributions.

This chapter is based on the following article:
F. Ruscelli, A. Laurenzi, E. Mingo Hoffman and N. G. Tsagarakis, "A Fail-Safe Semi-Centralized Impedance Controller: Validation on a Parallel Kinematics Ankle," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

4.1 Background on Impedance Control

Similarly to the intrinsic elasticity of human muscles, which allows small adaptations to external obstacles and partial energy absorption from impacts (Pratt et al., 2001), an effective solution to improve the interaction skills of robotic systems experiencing frequent dynamic interactions with the environment is the employment of impedance control schemes (Hun-ok Lim et al., 2001; Jong Hyeon Park and Hoam Chung, 1999). Interacting effectively with the environment is very relevant to humanoids and in general for loco-manipulation tasks where the ability to conform and adapt to ground disturbances and impacts is a fundamental prerequisite for achieving robustness. Impedance control constitutes an intuitive way to correlate nominal system trajectories with external stimuli. It establishes a dynamic relationship between the joint signals and the forces exerted by the environment on the system, providing a way to effectively adapt the robot behaviour based on external inputs.

The relationship between the joint angles $\boldsymbol{q} \in \mathbb{R}^N$ of a robot and the external forces $\boldsymbol{\tau}_{ext}$ is formulated as: specified by:

$$\boldsymbol{K}(\boldsymbol{q}_\mathrm{d} - \boldsymbol{q}) + \boldsymbol{D}(\dot{\boldsymbol{q}}_\mathrm{d} - \dot{\boldsymbol{q}}) + \boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_\mathrm{d} - \ddot{\boldsymbol{q}}) = \boldsymbol{\tau}_{ext} \tag{4.1}$$

The dynamical system defined is mono-stable, where the desired angles $\boldsymbol{q}_d$ vary around a single attractor, the joint position $\boldsymbol{q}$, in response to a forcing term $\boldsymbol{\tau}_{ext}$. Indeed, the imposed dynamic relationship between force and position, velocity and acceleration mimics a *spring-mass-damper* behavior on the mechanism: the matrices $\boldsymbol{M}$, $\boldsymbol{B}$, and $\boldsymbol{K}$ are respectively effective mass, damping and spring constant matrices. Note how the spring-mass-damper stores and dissipates the energy coming from an external impact. Hence, via the impedance control, it is possible to control the energy exchange during an interaction of the robot with the environment. Adding a middle-layer of impedance control allows defining a control architecture that naturally includes compliance for articulated locomotion.

4.2 Contribution

In implementing any control strategy particular emphasis should be put on the robot reliability: reducing, in the presence of communication interruptions between the centralized high layer controllers and the local actuation regulators, the risk of abrupt and unwanted reactions that could damage the robot and the environment or create unsafe conditions in human-robot interaction tasks. However, one of the limitations of the classical impedance control is its sensitivity to interruptions of torque reference signals for the local joint torque controllers. Due to the inherent instability of a pure joint torque regulator, if the communication between the centralized impedance control and the joint torque controllers is interrupted, unwanted torques are commanded to the motors. This could compromise the overall robot self-stability, creating dangerous situations for humans interacting with the robot and increasing the risk of damages both to the physical system and the environment.

The contribution of the work presented in this chapter consists in realizing the desired impedance using a partially centralized scheme that reduces abrupt motions when the torque references of the actuators are interrupted. Such events could lead to unwanted torques commanded abruptly to the robot, leading to hardware damages or sudden motions, especially dangerous in human-robot interaction tasks. Specifically, the work is divided into two parts, both oriented towards the goal of devising an impedance controller capable of partially

rejecting external perturbation while being more stable against interruptions in the actuation reference signals:

- the implementation of an impedance controller for the robot COMAN+;
- the design of the Semi-Centralized Impedance Controller (SCIC), a specific formulation of the impedance controller based on a semi-centralized scheme, to ensure the system stability and safety in the presence of failures (see Figure 4.2).

The most critical part of the robot concerning safety and compliance has been identified with the ankles: the proposed control is implemented on the ankle joint, which is based on a parallel kinematics mechanism (Zhou and Tsagarakis, 2018) and actuated by two motors through a dual four-bar transmission mechanism. Thereby, an improvement of the humanoid's balance is expected: on one side it is granted by the ability to adjust the ankles' position in response to external perturbations of the foot, while on the other it is ensured by the energy absorption from collisions, which mimics the intrinsic elasticity of human muscles. Since the foot is the joint directly interacting with the terrain and hence subjected to impact forces and disturbances from the ground, compliant ankles could enhance the robot behaviour on uneven grounds, adapting its step to the terrain and rejecting minor disturbances on the joint. The main motivation of this work is the realization and validation of the aforementioned controller, to improve robot reliability: ensuring system stability and safety in the presence of failures, i.e. communication interruptions between the on-board centralized computer and actuation Digital Signal Processor (DSP) controllers. Such events could lead to unwanted torques commanded abruptly to the robot, leading to hardware damages or sudden motions.

The SCIC takes advantage of the local joint impedance controller implemented at each motor's DSP, sending the desired torque as two different components: the first depends only on the reference and the sensed position of the motor, while the second term is a bounded torque sent in feed-forward. This allows the robot, if the communication is interrupted, to keep the last motor position reference sent, while the feed-forward term becomes a bounded disturbance which is easily rejected. In the next sections, an in-depth description of the transmission mechanism between joints and motors of the ankle is given, followed by the introduction of the semi-centralized impedance controller. The proposed controller is validated on the ankles of COMAN+ and experimental results and the relative discussion are provided.

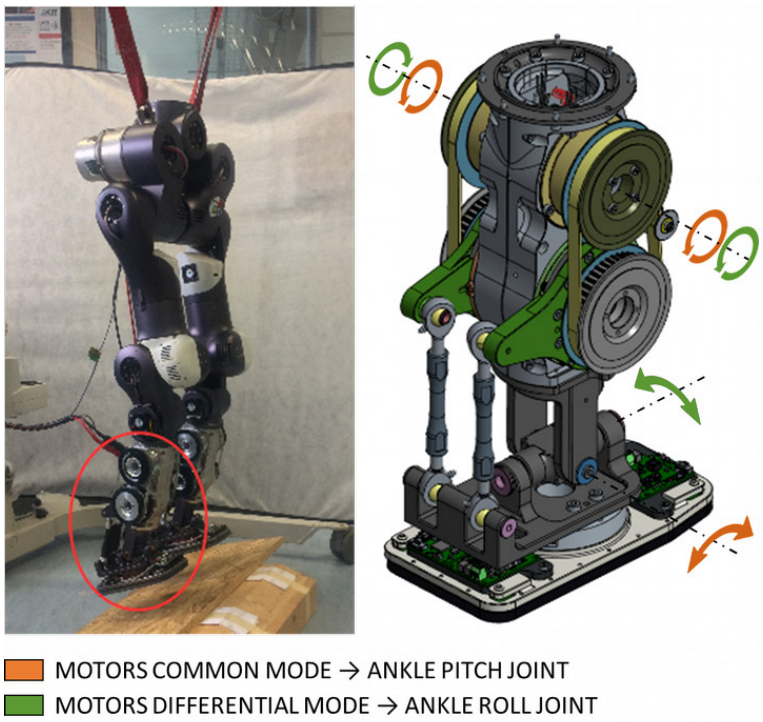


Figure 4.1 The legs of the humanoid robot COMAN+ on the left and close back view of its tibia (CAD) without covers on the right.

4.3 Joint-to-motor transmission

In this section, the parallel ankle mechanism and the kinematics of the joint-motor transmission is described. A parallel ankle offers numerous advantages over its serial counterpart, though at the expense of more complex kinematics. By using a four-bar mechanism the motors of the ankle can be placed on the shin, decreasing the inertia of the tibia, improving the dynamic behaviour and reducing power consumption. Additionally, a parallel configuration allows the distribution of the torque provided by the actuators to the two DoFs of each ankle, shown in Figure 4.1: when the motors situated on the shin move in common mode (both rotate in the same direction, as illustrated by the orange arrows), a pitch motion is produced. Conversely, the motors moving in differential mode result in a roll motion of the ankle (green arrows). The transmission between the parallel ankle joint and actuators requires a transformation when passing from the motors variables to joint variables. In particular, joint and motor angles are related through a forward kinematic mapping, and differentiation yields the velocity mapping:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \tag{4.2}$$

where $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{2\times 2}$ is the Jacobian for the parallel ankle mechanism of COMAN+, $\dot{\boldsymbol{q}} \in \mathbb{R}^2$ are the joint-side velocities and $\dot{\boldsymbol{\theta}} \in \mathbb{R}^2$ the motor-side velocities. A complete computation of the Jacobian for this mechanism can be found in Zhou and Tsagarakis (2018). From the principle of virtual work, a linear mapping between the motor and joint torques is obtained, which is required for the design of the ankle impedance controller:

$$\boldsymbol{\tau}_m = \boldsymbol{J}(\boldsymbol{\theta})^\top \boldsymbol{\tau}_j, \tag{4.3}$$

where $\boldsymbol{\tau}_m \in \mathbb{R}^2$ is the torque sensed by the actuator and $\boldsymbol{\tau}_j \in \mathbb{R}^2$ is the torque at the joint (from now on the dependency on $\boldsymbol{\theta}$ is dropped).

4.4 Impedance controller

In this section, it is described the implementation of an impedance controller on the parallel kinematics ankles of the humanoid robot COMAN+. The impedance controller at the motor-side has the following formulation:

$$\boldsymbol{\tau}_{m,d} = \mathbf{K}_m \Delta\boldsymbol{\theta} + \mathbf{D}_m \Delta\dot{\boldsymbol{\theta}} + \boldsymbol{\tau}_{m,\text{ff}}, \tag{4.4}$$

where $\mathbf{K}_m \in \mathbb{R}^{2\times 2}$ and $\mathbf{D}_m \in \mathbb{R}^{2\times 2}$ are respectively the motor-side stiffness and the damping matrices, $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_d - \boldsymbol{\theta} \in \mathbb{R}^2$ is the error between the desired and measured motor positions, $\Delta\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{\theta}}_d - \dot{\boldsymbol{\theta}} \in \mathbb{R}^2$ is the error between the desired and measured motor velocities and $\boldsymbol{\tau}_{m,\text{ff}} \in \mathbb{R}^2$ are feed-forward reference torques at the motor-side. The same law can be defined at the ankle joint level:

$$\boldsymbol{\tau}_{j,d} = \mathbf{K}_j \Delta\boldsymbol{q} + \mathbf{D}_j \Delta\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{j,\text{ff}}, \tag{4.5}$$

where the quantities are defined w.r.t. the joint-side instead.

The goal is to define a motor-side stiffness and damping matrix corresponding to the desired joint-side stiffness matrix:

$$\{\mathbf{K}_m, \mathbf{D}_m\} = \mathcal{T}\{\mathbf{K}_j, \mathbf{D}_j\}, \tag{4.6}$$

Since the ankle has two DoFs, pitch and roll, suitable diagonal stiffness and damping matrices are designed to decouple the effect of the impedance controller on the two axes:

$$\mathbf{K}_j = \begin{pmatrix} k_p & 0 \\ 0 & k_r \end{pmatrix}, \tag{4.7}$$

By using (4.3) and (4.4) it is possible to write:

$$\begin{aligned}\mathbf{K}_m = \frac{\partial \boldsymbol{\tau}_m}{\partial \boldsymbol{\theta}} &= \frac{\partial (\boldsymbol{J}^\top \mathbf{K}_j \Delta \boldsymbol{q})}{\partial \boldsymbol{\theta}} \\ &= \boldsymbol{J}^\top \mathbf{K}_j \boldsymbol{J} + \cancelto{\mathbf{0}}{\frac{\partial \boldsymbol{J}^\top}{\partial \boldsymbol{\theta}} \mathbf{K}_j \Delta \boldsymbol{q}}.\end{aligned} \tag{4.8}$$

It can be seen how this formulation is position-dependent, since the mapping $\mathcal{T}$ changes with the robot configuration. If the stiffness matrix is computed around the equilibrium point ($\Delta\boldsymbol{\theta} = \mathbf{0}$), the second term $\frac{\partial \boldsymbol{J}^\top}{\partial \boldsymbol{\theta}} \mathbf{K}_j \Delta \boldsymbol{q}$ can be discarded, though neglecting the conservativeness of the mapping: an in-depth analysis including this term is found in Albu-Schäffer and Hirzinger (2002). A similar approach can be used to obtain the damping $\mathbf{D}_j$:

$$\begin{aligned}\mathbf{D}_m = \frac{\partial \boldsymbol{\tau}_m}{\partial \dot{\boldsymbol{\theta}}} &= \frac{\partial (\boldsymbol{J}^\top \mathbf{D}_j \Delta \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{\theta}}} \\ &= \boldsymbol{J}^\top \mathbf{D}_j \frac{\partial \dot{\boldsymbol{q}}}{\partial \dot{\boldsymbol{\theta}}} = \boldsymbol{J}^\top \mathbf{D}_j \boldsymbol{J},\end{aligned} \tag{4.9}$$

$$\begin{aligned}\mathbf{K}_j = \begin{pmatrix} k_p & 0 \\ 0 & k_r \end{pmatrix}, \qquad \mathbf{D}_j = \begin{pmatrix} d_p & 0 \\ 0 & d_r \end{pmatrix}, \\ \mathbf{K}_m = \begin{pmatrix} k_a & k_c \\ k_c & k_b \end{pmatrix}, \qquad \mathbf{D}_m = \begin{pmatrix} d_a & d_c \\ d_c & d_b \end{pmatrix}.\end{aligned} \tag{4.10}$$

In general, from a diagonal joint stiffness matrix $\mathbf{K}_j$, a non-diagonal motor stiffness matrix $\mathbf{K}_m$ is derived from the mapping. Finally, the term $\boldsymbol{\tau}_{m,\text{ff}}$ can be obtained from $\boldsymbol{\tau}_{j,\text{ff}}$ using (4.3).

It should be noted that $\frac{\partial (\boldsymbol{J}^\top \mathbf{K}_j \Delta \boldsymbol{q})}{\partial \boldsymbol{\theta}} = \boldsymbol{J}^\top \mathbf{K}_j \boldsymbol{J}$ is valid only locally (Albu-Schäffer et al., 2004; Albu-Schäffer and Hirzinger, 2002). Therefore, large displacements from the equilibrium point lead to errors in the stiffness matrix. However, since the parallel ankle mechanism has a narrow range of motion, both on the pitch and the roll axes (i.e. the term $\Delta \boldsymbol{q}$ always remains relatively small), the mapping never causes significant errors.

Remark: In this chapter, a traditional notation was used to describe the motor-joint mapping $\boldsymbol{q} = f(\boldsymbol{\theta})$ for the sake of clarity. However, in the robot the inverse mapping was used, since the kinematics were provided in this form in Zhou and Tsagarakis (2018). Following the calculations, the inverse of the Jacobian is required to obtain the motor torque: $\boldsymbol{\tau}_m = \boldsymbol{J}(\boldsymbol{\theta})^{-T} \boldsymbol{\tau}_j$. Thanks to the mechanical design of the ankle mechanism the matrix is square and has always full rank: therefore, this prevent to run up against singularity issues.

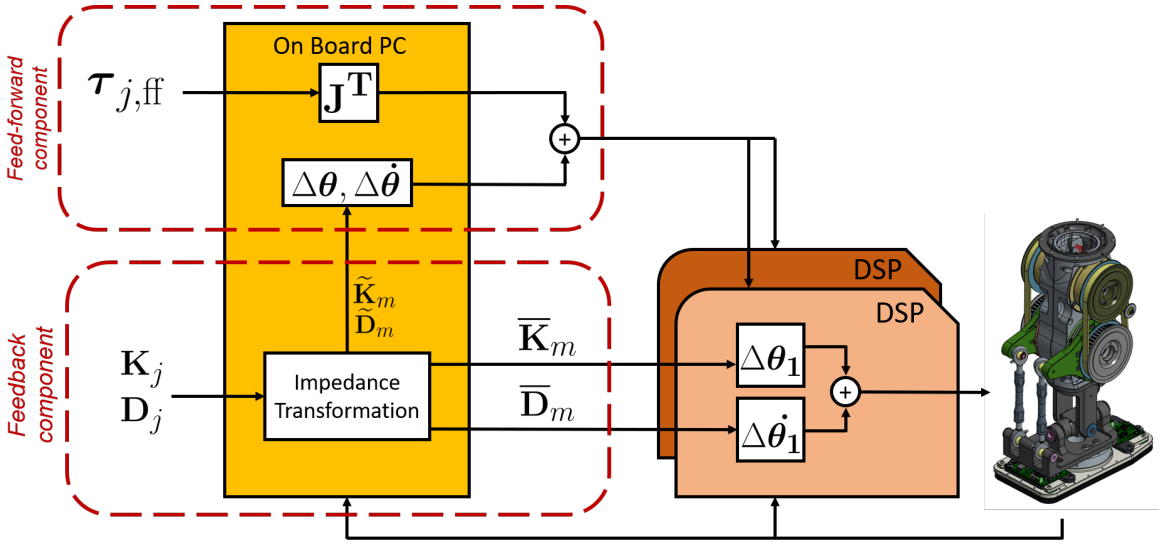


Figure 4.2 Scheme of the semi-centralized impedance controller (SCIC).

4.5 Semi-Centralized Impedance Control (SCIC)

In this section is presented a novel implementation of an impedance controller to make the ankles less susceptible to hardware breakdown or disconnections, reducing potential movements that could compromise the hardware or create unsafe conditions in general. To achieve the desired behaviour in the ankle joint, the controller can be implemented in two different ways for a torque-controlled robot.

Pure Torque Control. A conventional way is sending the computed torque directly to the torque controller implemented in each DSP. The on-board processor computes joint torques and transforms them into motor torques before sending them to the DSP of the actuator:

$$\boldsymbol{\tau}_{m,d} = \boldsymbol{J}^{\top}\left(\mathbf{K}_j\Delta\boldsymbol{q} + \mathbf{D}_j\Delta\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{j,\mathrm{ff}}\right). \tag{4.11}$$

This method, although viable, is not robust against hardware failures that cause interruptions of the torque reference signals sent to the joint local torque controllers. If the connection between the computer and the DSP is severed, the robot behaviour can become unstable, generating high-speed motions that could damage the robot and the environment: having solely a torque reference and no position feedback, this impedance control implementation would directly send the last computed torque reference to the DSP causing the ankle to jolt against the mechanical limits with the risk of damaging the actuator transmission system or other mechanical components. Besides, such behaviour can severely compromise safety in any robotics application where humans are in close vicinity of the robot. Since most of

the human-robot collaboration applications rely on the use of impedance control to generate compliant interactions, bounding the robot reactions in the case of system failures is critically necessary.

Semi-Centralized Impedance Control. A way to grant a safer response is sending to the DSP a position reference and the computed stiffness and damping diagonals, and let the DSP calculate the torque using the feedback of the robot. This allows bounding the system motion reaction, resulting in a graceful and safer degradation if a system failure causing discontinuities of the torque reference signals occurs.

To achieve this scheme (shown in Figure 4.2), the matrices $\mathbf{K}_m$ and $\mathbf{D}_m$ can be arranged in the following way, derived from the joint-motor transformation:

$$\begin{aligned} \mathbf{K}_m &= \overline{\mathbf{K}}_m + \widetilde{\mathbf{K}}_m, \\ \mathbf{D}_m &= \overline{\mathbf{D}}_m + \widetilde{\mathbf{D}}_m, \end{aligned} \tag{4.12}$$

where $\overline{(\cdot)}_m$ contains the diagonal elements and $\widetilde{(\cdot)}_m$ contains the off-diagonal elements of the matrix. Recalling equation (4.4):

$$\boldsymbol{\tau}_{m,d} = \overline{\mathbf{K}}_m \Delta\boldsymbol{\theta} + \overline{\mathbf{D}}_m \Delta\dot{\boldsymbol{\theta}} + \boldsymbol{\tau}_{m,\mathrm{ff}}, \tag{4.13}$$

where:

$$\boldsymbol{\tau}_{m,\mathrm{ff}} = \widetilde{\mathbf{K}}_m \Delta\boldsymbol{\theta} + \widetilde{\mathbf{D}}_m \Delta\dot{\boldsymbol{\theta}} + \boldsymbol{J}^\top \boldsymbol{\tau}_{j,\mathrm{ff}}. \tag{4.14}$$

This formulation enables the decentralization of the input signals. The first two terms in (4.13) are computed in each DSP, each row separately. Conversely, the $\boldsymbol{\tau}_{m,\mathrm{ff}}$ component in (4.14) is computed on the on-board computer in a centralized way: since the i^{th} DSP has information solely about the i^{th} actuator to which is connected, it can only determine the torques dependent on position θ_i and velocity $\dot{\theta}_i$, whereas both terms (θ_i and $\dot{\theta}_i$, $\forall i = 1,2$) are needed to obtain each motor total torque. Therefore, the on-board computer uses the off-diagonal terms $\widetilde{\mathbf{K}}_m$ and $\widetilde{\mathbf{D}}_m$ to compute the part of the torque that depends on the other joint, which is then sent directly to the DSP as a feed-forward component. Term $\boldsymbol{\tau}_{j,\mathrm{ff}}$ is a torque directly commanded to the joint by the user. Thus, in case of a failure:

- the last position sent to the robot is tracked by the DSP which continuously receive the position feedback locally;
- the last feed-forward torques received, provided by the remaining components of the stiffness and damping matrices, will act as a disturbance in the resultant position feed-

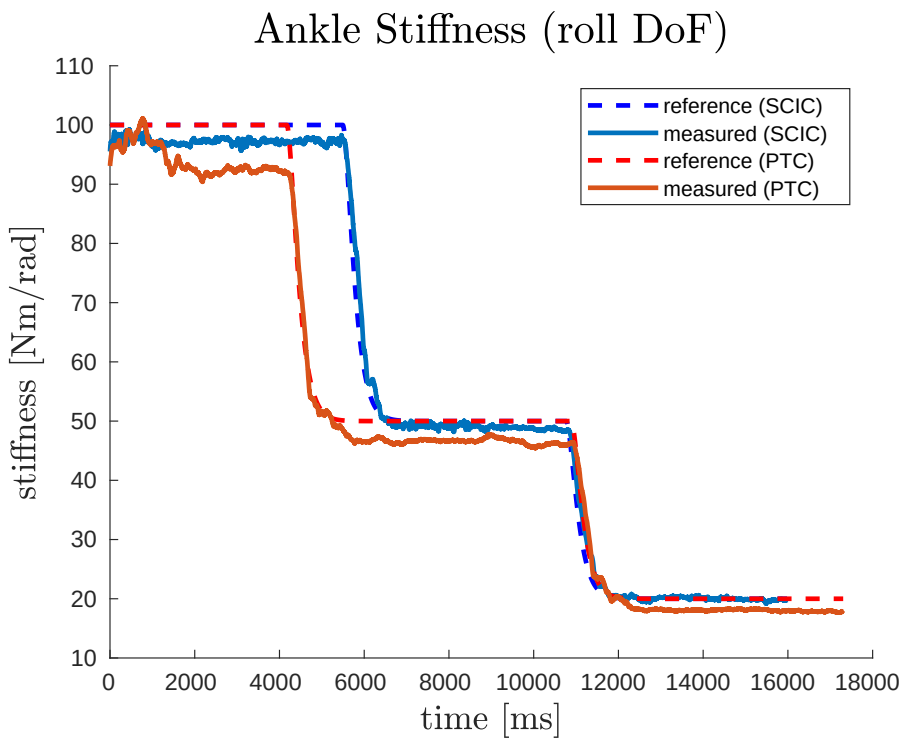


Figure 4.3 Reference stiffness (*roll* axis) commanded manually (steps of 100, 50, 20 Nm/rad) compared to measured ankle stiffness with SCIC and PTC.

back loop implemented using the diagonal stiffness and damping elements. However, this disturbance is always bounded and not large enough to induce instabilities.

The conceived controller, in case of system faults that could interrupt the reference signals to the joint local controllers, is then reduced to a simple closed loop with a bounded disturbance at the input which can be rejected. This favors the hardware protection and results in slower and safer robot reactions, especially important in human-robot interaction.

Remark: A similar semi-centralized method was implemented in Albu-Schäffer and Hirzinger (2002) under different circumstances, i.e. a Cartesian impedance controller used to implement a compliant behaviour for robotic manipulators. However, the rationale behind this implementation differs from the one in Albu-Schäffer and Hirzinger (2002), as in this project it is exploited to demonstrate its benefits in ensuring a bounded robot reaction in case of system failures.

4.6 Experiments

The robot platform used in this work is the lower body of the humanoid robot COMAN+. As presented in Chapter 3, it is composed of 13 series elastic actuated (SEA) joints, providing a first layer of passive compliance and the ability to sense external forces. The ankles motors were controlled in impedance mode: the control loop in each DSP consists of an impedance controller which takes as inputs the desired torque together with desired values of stiffness and damping. The impedance controller sends the computed reference torque to the lower-level torque controller of the actuator.

To validate it, the SCIC is compared to the conventional pure torque control (PTC). In particular, two classes of experiments are designed to estimate performance, detailed in Figure 4.6.1 and 4.6.2: the first one is a set of experiments to measure the accuracy of the SCIC in rendering the desired impedance, and the second is a qualitative comparison to prove its efficacy in reducing the system motion reaction when a failure occurs. Finally, an experiment was designed to show qualitatively the advantages of an impedance controller implemented on the ankle joint. While lifted by the crane, the robot was set in a fixed standing configuration. Subsequently, it was lowered on an inclined wooden board. The experiment was repeated using three different strategies on the ankle joint, as shown in Figure 4.9:

- impedance control mode;
- position control mode;
- no control.

Videos of the experimental comparisons are provided[1].

4.6.1 Accuracy validation

For the sake of clarity, two independent set of experiment are conducted, one for the stiffness alone and one for the damping, so that the effects of damping and stiffness in (4.5) can be decoupled. The feed-forward commanded torque was set to zero. Separate experiments were carried out for both ankle DoFs. The robot was lifted with a crane, raising it above the ground, starting with a nominal commanded stiffness (or damping). Then, an external force was applied along one DoF of the left ankle, while the commanded stiffness (or damping) was changed by the user. In particular, the ankle was kept in place while changing the stiffness reference, since its value depends on the position error. On the contrary, the damping depends

[1] https://youtu.be/-QULP4xUD9A

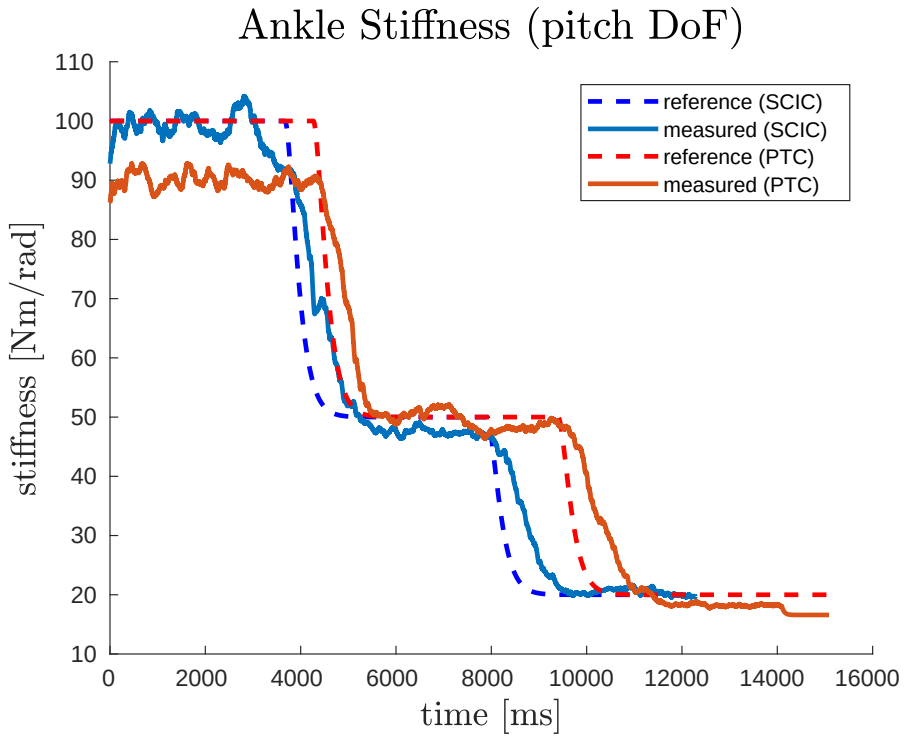


Figure 4.4 Reference stiffness (*pitch* axis) commanded manually (steps of 100, 50, 20 Nm/rad) compared to measured ankle stiffness with SCIC and PTC.

on the velocity error; hence the ankle was moved while changing the damping reference to estimate its value. The sensed motor torque and position were translated into the sensed joint torque and position by using the Jacobian associated with the joint-motor mapping. A simple way to obtain the measured stiffness (or damping) is rearranging (4.5), but the results are way too noisy. Hence, it was used a Recursive Least Square (RLS) filter to estimate the desired values. This adaptive algorithm is used to recursively evaluate stiffness and damping over time by minimizing a weighted least square cost function. Two experiments are conducted for each degree of freedom, one for the stiffness and one for the damping.

4.6.2 Efficacy validation

To validate the proposed controller compared to the traditional one in the case of communication interruptions, the same experimental setting of the previous experiments is used, keeping the robot above the ground. While the SCIC was controlling the ankle with a non-zero stiffness and damping, a force was applied along one axis. In this configuration, the Ether-

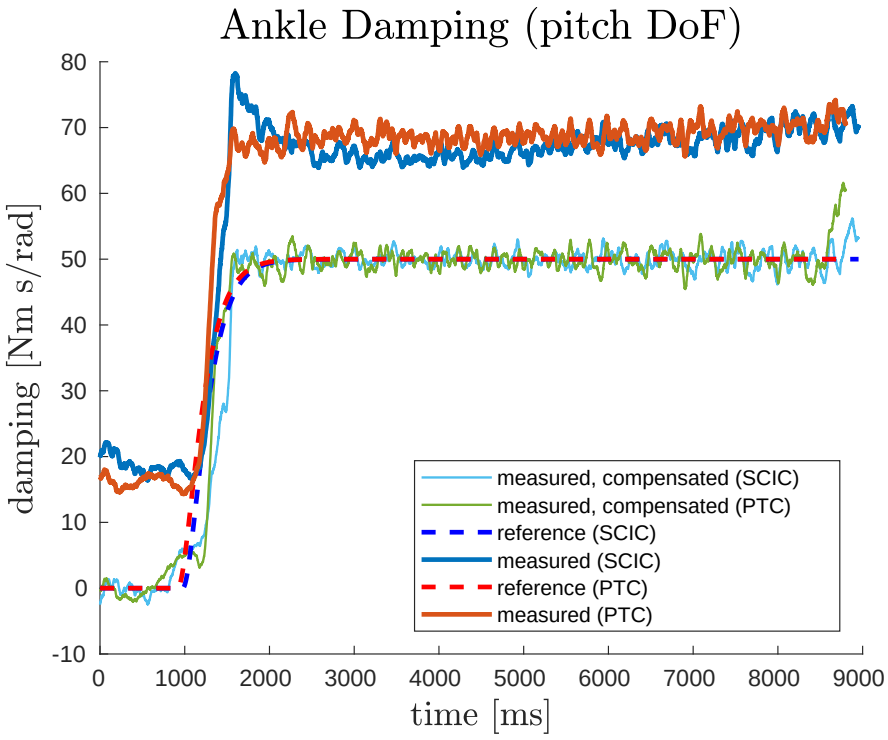


Figure 4.5 Reference damping (*pitch* axis) commanded manually (one step from 0 to 50 Nm s/rad) compared to measured ankle damping with SCIC and PTC. The offset between measured and reference values is due to a torque tracking error. The compensated measures are shown, closely tracking the reference values of damping.

CAT cable connecting the on-board computer to the actuators was unplugged, simulating a hardware failure. The same experiment was conducted with a pure torque input controller.

4.6.3 Results

The first class of experiments validated the accuracy of this method compared to the traditional torque input approach. Figures 4.3 and 4.4 shows the reference stiffness signal and the measured stiffness using the SCIC and a pure torque input (PTC) for the roll and pitch DoF, respectively. Figures 4.5 and 4.6 shows the same graphs for the measured damping. All the measured values are filtered with the RLS filter. Note that the differences between the two reference values in each graph are due to the fact that the data for the SCIC and the PTC belong to separate experiments and the commands sent to the robot by the user differ in time. The measures from the SCIC are close to the reference signal and comparable to the ones from the PTC, thus validating the proposed method.

The second experiment demonstrates the effectiveness of the SCIC's design: as the disconnection between the on-board computer and DSPs occurs, the ankle's DSP keeps the last position sent by the SCIC, while using the traditional control the last torque sent before the disconnection is kept as the reference input, resulting in a rapid motion of the ankle joint towards its limit. Figure 4.8 shows a sequence of frames comparing the behaviour of the robot controlled with the SCIC and the PTC. Due to the nature of this particular experiment, i.e. unplugging the etherCAT cable to simulate a hardware failure, it's not possible to collect any data.

Finally, the qualitative experiment proved the importance of impedance control in granting the robot ankles the ability to conform to the terrain. The ankles adapt to the slope without yielding under the weight of the robot. This desired behaviour falls between two unwanted extremes, shown in the remaining trials of Figure 4.9. In a) the robot is position-controlled. The ankles do not adapt to the slope, making the robot fall backward. In b) the robot is controlled with an impedance controller (stiffness: 30 Nm/rad, damping: 5 Nm s/rad), which allows the ankles to comply with the inclined board. Finally, in c) the robot is not controlled. The ankles collapse under the weight of the robot, making it fall forward. The stiff position control does not allow the ankle to comply with the terrain, which rigidly keeps its position when the robot is lowered on a wooden board (inclined of 16°), while a non-controlled ankle collapses under the weight of the robot. Note that in these experiments it is not implemented any balancing strategy on the robot, which is standing entirely thanks to its initial configuration. A video including the full sequence of the controllers' comparison (SCIC-PTC) and the qualitative experiments is made available[2].

4.6.4 Offset analysis

In this section, the offset between the reference signal and the measured values in the graphs of Figures 4.3, 4.4, 4.6 and 4.5 is discussed. The offset error, which is particularly evident in Figure 4.5, arises from an offset in the torque tracking of the motor lower level control, which is shown in Figure 4.7. The corresponding stiffness and damping offset can be computed by solving the equation (4.5) for the stiffness or the damping. Note that the equation always has one addend only, since in the experiments either pure stiffness or damping was commanded to the robot and the feed-forward component was set to zero. The following expression is

[2]https://youtu.be/-QULP4xUD9A

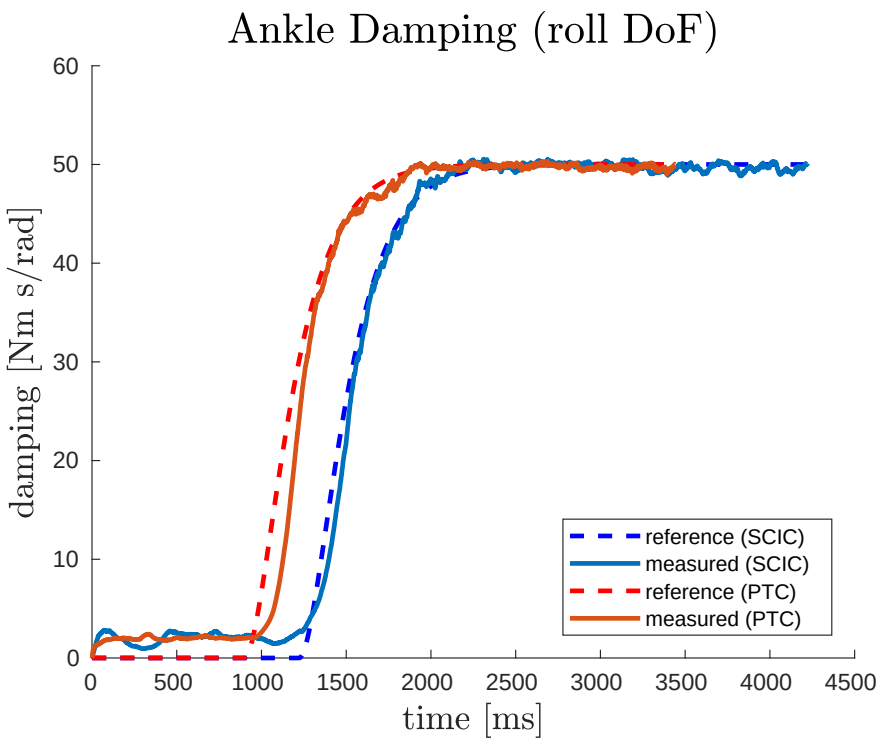


Figure 4.6 Reference damping (*roll* axis) commanded manually (one step from 0 to 50 Nm s/rad) compared to measured ankle damping with SCIC and PTC.

obtained for the stiffness offset:

$$\delta \boldsymbol{K}_j^{\text{meas}} = \text{diag}\,(\Delta \boldsymbol{q})^{-1}\, \boldsymbol{J}^{-T} \cdot \delta \boldsymbol{\tau}_j^{\text{meas}} \tag{4.15}$$

where $\delta \boldsymbol{K}_j^{\text{meas}}$ and $\delta \boldsymbol{\tau}_j^{\text{meas}}$ are the offset error in the measured stiffness and torque, respectively. Since the torque on the motor level is measured, $\delta \boldsymbol{\tau}_j^{\text{meas}}$ can be computed mapping the sensed torque to the joint level, using equation (4.3).

A similar equation can be written to estimate the offset in the damping:

$$\delta \boldsymbol{D}_j^{\text{meas}} = \text{diag}\,(\Delta \dot{\boldsymbol{q}})^{-1}\, \boldsymbol{J}^{-T} \cdot \delta \boldsymbol{\tau}_j^{\text{meas}} \tag{4.16}$$

Both experiments to estimate stiffness and damping have an offset, but in the stiffness experiments the error is significantly smaller. This is due to the different nature of the two experiments: since the quality of the torque tracking is worse during torque reference variations, the experiment to estimate the damping, in which the ankle is moved, is affected by a higher torque tracking error compared to the experiment to estimate the stiffness, in

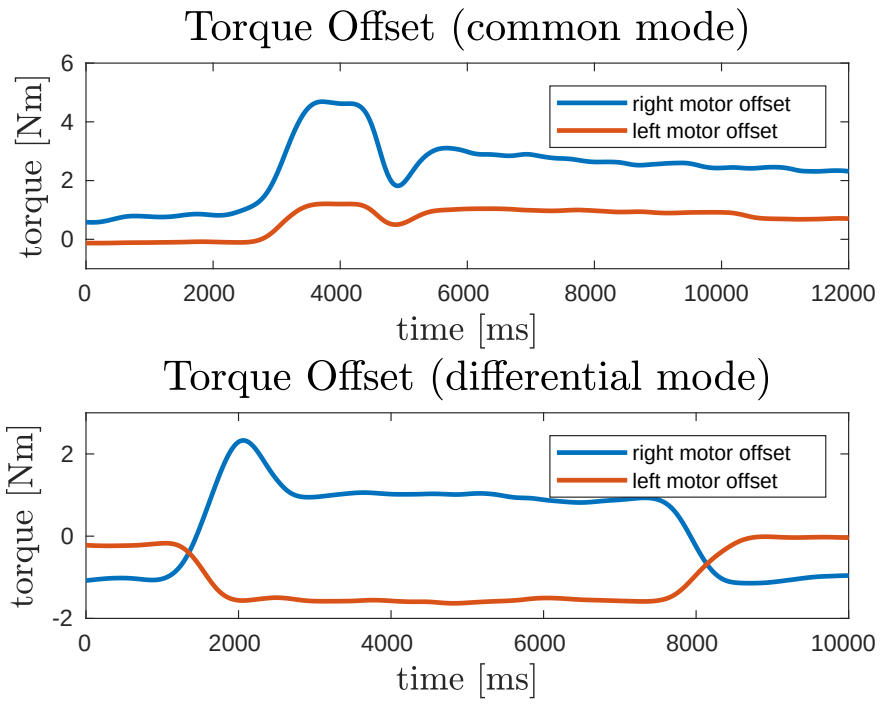
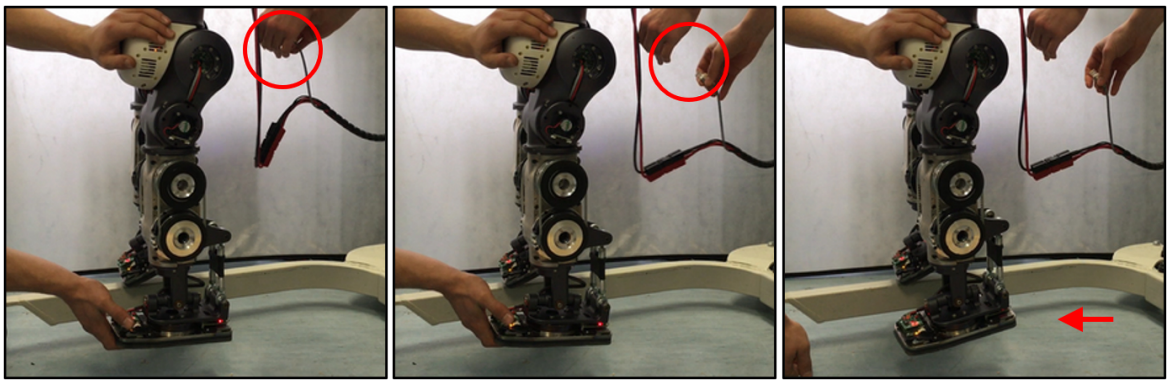


Figure 4.7 Torque offset caused by a torque tracking error of the motor lower-level control, filtered with a zero-phase digital filter. Above: the ankle motors are moving in common mode, resulting in a rotation of the ankle along the pitch axis. The torque offsets have the same sign. Below: the ankle motors are moving in differential mode, resulting in a rotation of the ankle along the roll axis. The torque offsets have opposite sign.

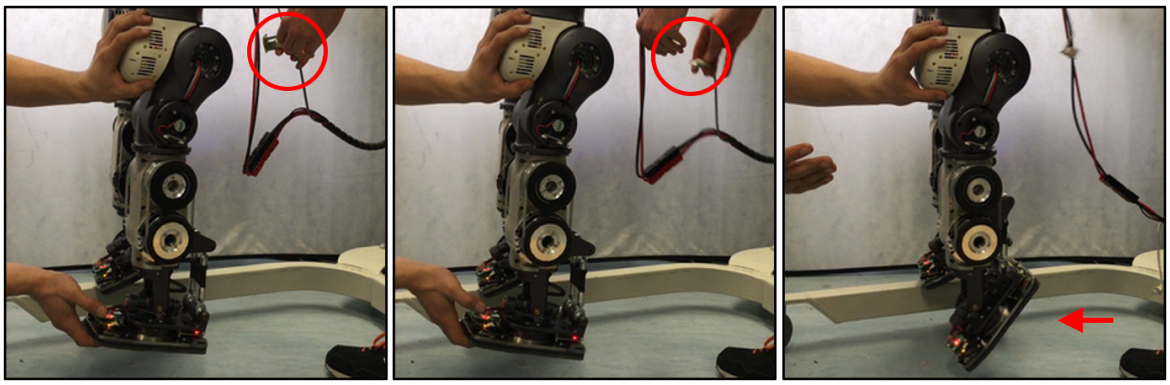
which the ankle stays still. In the first graph of Figure 4.7 the two motors actuating the ankle joint are moving in common mode, and their offsets are concordant. In the second graph, the two motors are moving in differential mode, and their offsets are discordant. This justifies both the offset in Figure 4.5, where the offsets from the two motor readings sum up, as well as the absence of an error in Figure 4.6, where the two errors cancel each other out. Finally, Figure 4.5 shows the damping measures corrected by compensating the offset: the resulting values follow closely the reference signal.

4.7 Discussion

The semi-centralized impedance controller was successfully implemented on the COMAN+ robot, extending it for the parallel kinematics mechanism of the ankles. It was experimentally demonstrated the effectiveness of this method, which performs closely to a conventional



(a) SCIC allows a safe response of the robot when the external force exerted on the robot is removed, the robot tracks the last position sent by the controller, returning to the nominal configuration.



(b) The PTC makes the ankle jolt, reaching violently the joint limit, when the external force exerted on the robot is removed. This sudden reaction could push the robot off balance or cause damages to the ankle joint.

Figure 4.8 Sequence of frames of the robot COMAN+'s behaviour when disconnected from the on-board computer. The red circles highlight the etherCAT, which is unplugged in the second frame simulating a hardware failure.

impedance control implementation based on pure local torque controllers: no evident difference is present between the experimental data of the SCIC and the PTC, as shown in Figures 4.3 - 4.6. The offset error in the measurements, due to the torque tracking error in the lower control, was analyzed and effectively compensated. Furthermore, it is showed how the implementation of the impedance controller improves the behaviour of the COMAN+ lower body, allowing the ankles to comply with the environment.

The main contribution of this chapter is the design, implementation and testing of an impedance controller capable of withstanding potential system failures without generating rapid motion reactions, which could compromise the hardware integrity or the safety of humans interacting with the robot. This is a fundamental prerequisite for a robotic platform used in real-time operations such as human-robot interactions or rescue missions, where

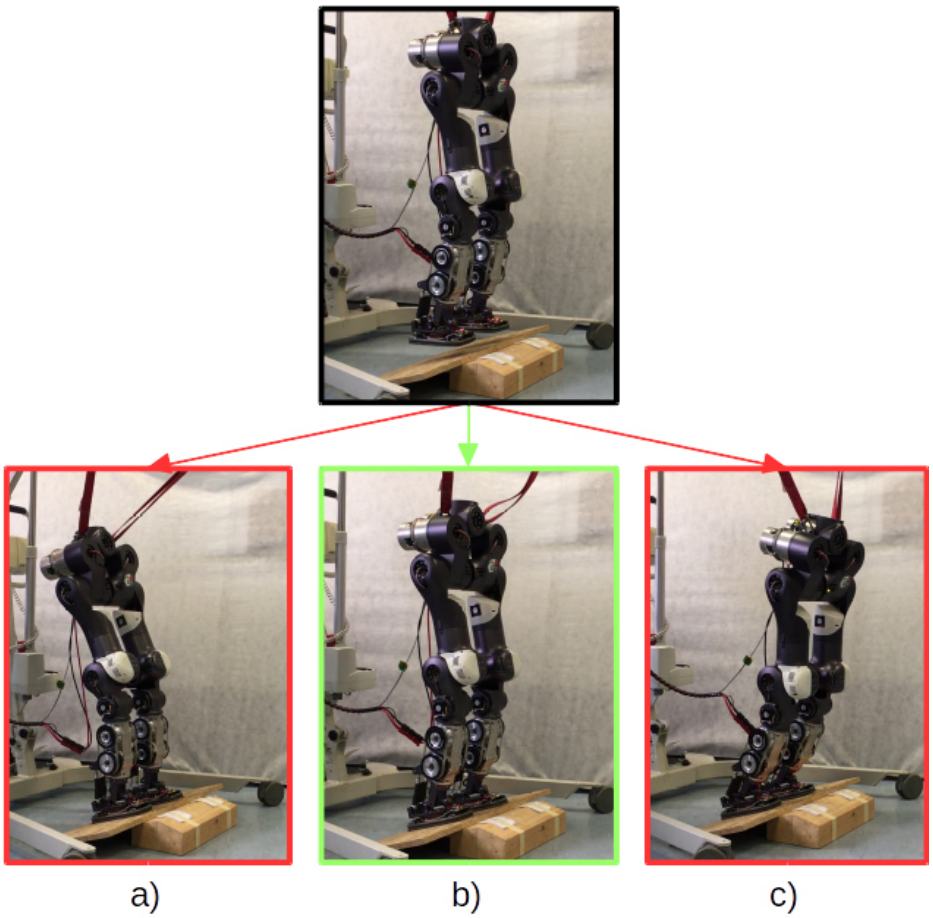


Figure 4.9 Frames of the COMAN+ robot being lowered on a inclined wooden board. Wrong behaviours are framed in red: a) Position control b) Impedance control c) No control.

the effects of failure should be reduced as much as possible to protect both the hardware and the human operators. The proposed implementation allows the robot to fail gracefully, as opposed to a traditional impedance control implementation relying on pure local torque controllers, in which interruptions or breakdowns of the main on-board computer could lead to abrupt movements of the joints and dangerous behaviours. The SCIC was experimentally validated in the presence of disconnections: simulating a hardware failure proves how the ankle does not drift violently, but returns to its nominal position. This project follows the development of the new robot COMAN+: future work will build upon this work to further develop and apply the proposed scheme to the full humanoid robot.

Part III

High Behavioural Level Control

Chapter 5

Gait Synthesis

The most relevant tools for contact stability analysis of humanoid robots were presented in Chapter 2. Given a set of reasonable assumptions, these tools can be exploited for the synthesis of robot motions. In this chapter, a novel strategy for gait synthesis is presented, merging results from the *Preview Control theory* and the *Limit Cycle Walking*. This choice is motivated by findings in bio-mechanics that show how the dynamic motion of the human walk is mainly generated by the sagittal component of the stepping.

This chapter is based on the following article:
F. Ruscelli, A. Laurenzi, E. M. Hoffman and N. G. Tsagarakis, "Synchronizing Virtual Constraints and Preview Controller: a Walking Pattern Generator for the Humanoid Robot COMAN+," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019.

5.1 Introduction and related works

The well-established approach of preview control proposed in Kajita et al. (2003a), takes advantage of future information of the walking pattern to generate appropriate trajectories for the robot. This promising direction of research was pursued in humanoid literature: from similar works that enhance the model to further reduce the modeling error and generate a robust pattern locomotion (Shimmyo et al., 2013) to application that extends preview control for external perturbations (Nishiwaki and Kagami, 2006; Nishiwaki and Kagami, 2009) or uneven terrains (Kajita et al., 2006). This strategy was framed in an MPC formulation, which allowed to generate the ZMP trajectory on-line with the control effort as the cost function

and with the balance condition as a constraint (Wieber, 2006), incorporating automatic footsteps placement (Herdt et al., 2010b) or enhancing the robustness of the MPC by making it intrinsically stable (Scianca et al., 2016).

This tool for bipedal locomotion, while being successful, inherently relies on the ZMP criterion, which confines the gait in the quasi-static domain where the equilibrium is determined by the spatial configuration of the body and feet (Goswami, 1999). This often results in unnatural motions compared to the fluid human walk, which includes phases of controlled falling where the ZMP is on the edge of the support polygon. The condition to guarantee a stable gait entails that the ZMP should stay strictly inside the support polygon: a more dynamic walking emerges if the gait allows phases during which this condition is violated. This can also be observed in studies on human walking (Sardain and Bessonnet, 2004), where it is shown that the ZMP mainly travels in the sagittal direction and reaches the corresponding edge of the support polygon, whereas in the lateral direction it remains well inside the support polygon. Hence, the dynamical motion is mainly determined by the controlled fall in the forward direction. To address these hypotheses, the conservative preview control is chosen to generate the lateral motion while a more dynamical framework such as the virtual constraints is picked for the sagittal motion. An extensive analysis of VCs and their application in bipedal mechanisms were presented in Westervelt et al. (2007), focusing on the planar case. The underlying assumption in this method is that dynamical and energy-efficient gaits are not achievable with standard ZMP, time-based strategies. Instead, it describes a walking gait as a periodic motion that resets at each impact and aims at designing a limit cycle of each joint involved in the robot movement. This is achieved by constraining these links to a phasing variable, greatly simplifying the kinematics of a complex system, and thus, the design of a periodic orbit.

While being a viable method that yielded successful results, it is usually applied to robots with particular body structures. Additionally, VC-based approaches are usually restricted to the sagittal plane and are not trivially generalized to the 3D case, where hybrid invariant manifolds are more challenging to describe. Nonetheless, many techniques were developed to implement 3D walking on different robots (Da et al., 2016; Reher et al., 2016), most of them decoupling the walking gait into forward and lateral motion. Examples include the functional Routhian reduction introduced in Grizzle et al. (2014), or the self-synchronization exploiting the symmetry of the system, formalized in Razavi et al. (2015). The practical limitations of dynamic walkers reside in periodic motions with a small basin of attraction and extreme dependence on the accuracy of the model: while adjustments can be made to expand the capabilities of this strategy (Manchester et al., 2011), the resulting gait is not very

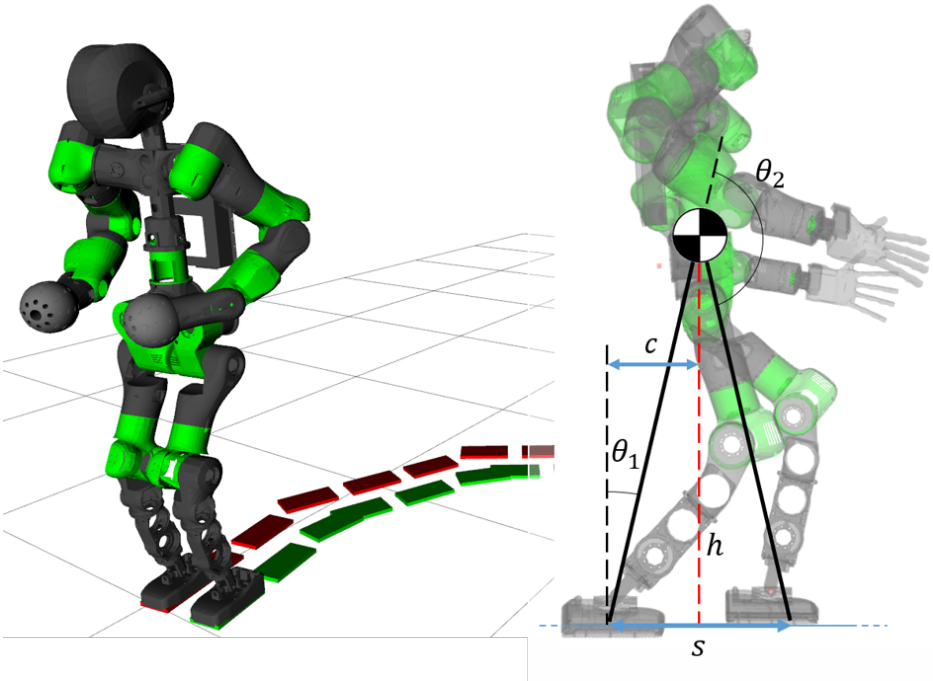


Figure 5.1 Left: the template model for the VC on the humanoid robot COMAN+ is the compass walker. θ_1 and θ_2 are the angles relative respectively to the *stance* leg and the *swing* leg. The CoM position c is always half of the step length s. Right: COMAN+ full kinematic model.

flexible, as noted in Rezazadeh et al. (2015). Furthermore, most of the strategies focus solely on straight walking gaits, due to the additional limitation introduced by a steering motion. In literature can be found some elegant solution, for instance in Motahar et al. (2017) and Chevallereau et al. (2010). However, they show mostly theoretical results supported only by simulation experiments.

5.2 Background

5.2.1 Preview Control

The seminal work in Kajita et al. (2003a) shows how, leveraging on the notion of the ZMP, a stable gait can be designed: this method outputs a feasible trajectory for the robot while keeping the advantages of a simple cart-table model. As discussed in Section 2.5.3, this model describes the ZMP trajectory as a function of the CoM motion. A walking pattern generation problem is the inverse of this: given a reference ZMP, the controller should output the CoM trajectory. By constructing a set of suitable dynamic systems, one could argue

that this problem can be solved as an output tracking control problem. However, as it is apparent from the graph in Figure 5.2, while the ZMP shifts from one reference value to the other (i.e. the robot switches the supporting leg after one full stepping motion) at time $t = 5$, the CoM starts moving at $t = 0$. The system is non-causal: since the CoM output trajectory moves *before* the ZMP input reference, effective tracking of the ZMP reference with an instantaneous control strategy is impossible. One possible solution is presented in Kajita et al. (2003a), which uses future information of the ZMP to generate a feasible CoM trajectory. From the ZMP-COM relationship of the cart-table equation 2.45, it defines two time-invariant linear discrete systems in the lateral and the sagittal plane, with the same form:

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \begin{bmatrix} 0 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_t + \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} \boldsymbol{u}_t \\ y &= \begin{bmatrix} 1 & 0 & -\frac{h_{\text{com}}}{g} \end{bmatrix} \boldsymbol{x}_t \end{aligned} \tag{5.1}$$

where the state vector is:

$$\boldsymbol{x} = \begin{bmatrix} x_{com} & \dot{x}_{com} & \ddot{x}_{com} \end{bmatrix}^\top \Big|_{x,y}, \tag{5.2}$$

and the control input $\boldsymbol{u} \in \mathbb{R}$ is the CoM jerk:

$$u = \frac{d}{dt} \ddot{x}_com \Big|_{x,y} \tag{5.3}$$

The output is the desired ZMP reference $y = x_{zmp}\Big|_{x,y} \in \mathbb{R}$. Assuming that at each time t in the future the values of $y_t, \ldots, u_{t+N-1}$ are available, the strategy consists in designing a optimal controller to minimize the tracking error $e_t = y_t - y_t^{ref}$, as well as the incremental state vector $\Delta \boldsymbol{x} = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$ and the incremental input $\Delta u = u_t - u_{t-1}$. Then, the *performance index* to minimize at each iteration is:

$$J = \sum_{k=t}^{\infty} Q_e \|e_t\|^2 + \|\Delta \boldsymbol{x}_t^\top Q_x \Delta \boldsymbol{x}_t\| + R\|\Delta u_t\|^2 \tag{5.4}$$

Notice how this approach includes the quadratic term for the rate of change in the state vector, which allows to directly regulate the transient of the state variables. By choosing a suitable future horizon length $N > 0$, the optimal controller u_k that minimizes the performance index (5.4) is:

$$u_t = -G_I e_t - G_S \boldsymbol{x} - \sum_{i=1}^{N} G_P(i) \boldsymbol{Y}_{ref} \tag{5.5}$$

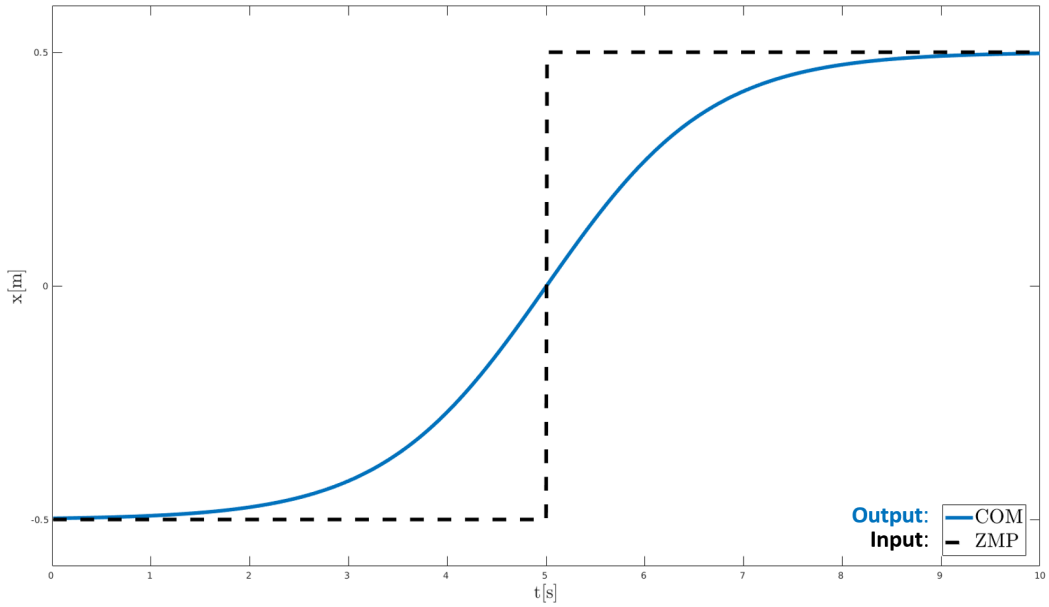


Figure 5.2 CoM trajectory (blue line) for a reference ZMP trajectory (black dotted line).

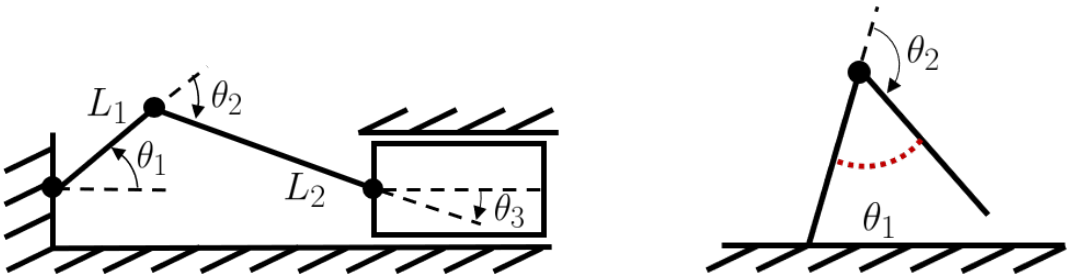
where $Y_{ref} \in \mathbb{R}^N$ is the prediction horizon that spans the future reference of the ZMP:

$$\boldsymbol{Y}_{ref} = [y_t, \ldots, y_{t+N-1}] \tag{5.6}$$

where G_I, G_S and G_P are the integral, state feedback and preview gain computed using the Riccati equation. A throughout description of their computation is found in Katayama et al. (1985).

5.2.2 Virtual Constraints

Controlling humanoid robots present many challenges directly stemming from the characteristics of these systems: as discussed in Chapter 2, under-actuation, contact conditions, hyper-redundancy and self-balancing aggravate the problem of designing a successful control architecture. Virtual constraints (VCs) greatly simplify the problem of robot control design as they reduce the dimensionality of a complex system linking together the evolution of different quantities. The concept of VCs is equivalent to their mechanical counterparts: to illustrate this, let's take a simple 2-link mechanism connected to a block between two walls, as in Figure 5.3a. Indeed, the block is constrained to move horizontally, and, as a result, it



(a) (b)

Figure 5.3 Comparison of a system limited by physical constraints due to the walls blocking limiting the mechanism motion (a) and one under the effect of VCs, represented with the red dotted line, induced by a controller (b). The resulting motion is constrained in both scenarios, even if in (b) there are no physical connections.

constrains the movement of the other two links:

$$\begin{aligned} 0 &= L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2) \\ \pi &= \theta_1 + \theta_2 + \theta_3 \end{aligned} \tag{5.7}$$

While the "free" mechanism has three DoFs, the constraints imposed by the walls reduce it to a 1-DoF system: by controlling one of the variables $\theta_1, \theta_2, \theta_3$, the whole system moves according to Equations (5.7). Indeed, physical constraints can be installed on a robot to bind together the motion of a set of desired joints: this approach is usually seen on passive walker, unactuated mechanism that locomote on slopes propelling themselves using their own weight. However, given proper actuation of the robot, the same result can be achieved by imposing a set of VCs:

$$\begin{aligned} y_1 &= L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2) \\ y_2 &= \theta_1 + \theta_2 + \theta_3 - \pi \end{aligned} \tag{5.8}$$

If the outputs y_1 and y_2 are kept to zero, the mechanical system will behave as if the block was constrained by the walls. The same strategy can be used on bipedal models, where the VC is designed to link together the motion of the two legs in order to obtain a periodic gait (see Figure 5.3b. The advantages of VCs over physical constraints lies in the flexibility of this method: the constraints linking the joints can be reconfigured at will, allowing for the design of different gaits.

5.3 Contribution

Motivated by observations in Sardain and Bessonnet (2004), the forward and the lateral motions of the walking are addressed separately, exploiting two different approaches: the VC for the sagittal plane and the preview control based on equality-constrained QP problem for the lateral plane. The main contribution of this work consists in merging the two approaches into an effective walking pattern generator for the humanoid robot COMAN+: the sagittal trajectory of the CoM, where the dynamical component of the gait is mainly found, is governed by a VC-based controller, to design a template periodic motion. Along the lateral plane, the classic PC is applied: the input-output relationship between CoM and ZMP of the one-dimensional linear pendulum is used to generate a CoM trajectory tracking the ZMP reference fed to the controller. Omni-directional gait synthesis based on the PC scheme is a solved problem, whereas the presented hybrid algorithm is not trivially generalizable for a flexible walk in different directions. The contribution of this work is the development of a novel hybrid WPG, which entails the design of a new set of VCs defined in the Cartesian space, and a method to guarantee synchronization between the lateral CoM swing and the phase-based swing trajectory of the step in the sagittal plane. The resulting WPG features:

- a lightweight and versatile architecture, it allows to change heading, step length and feet distance;
- a user interface, which enables to modify on-line the gait;
- a based on a simple template model, it can be easily deployed on any humanoid robot.

This method removes the limitation of the ZMP criterion along the sagittal direction without relinquishing the advantages of the PC. A whole-body IK solver maps the template model on the robot, by specifying a set of desired Cartesian tasks organized in a SoT fashion. This allows imposing the desired body posture while controlling the template model variables as high priority tasks, i.e. the feet and the CoM trajectories.

Using this hybrid strategy, it is investigated how the time-dependent preview control and the time-independent virtual constraints approach can be integrated together in humanoid locomotion. The results are be discussed in terms of advantages and limitations of this approach. The WPG is validated on the robot COMAN+ with different walking experiments, showing the effectiveness of this promising strategy. Finally, a perspective on future improvements is given.

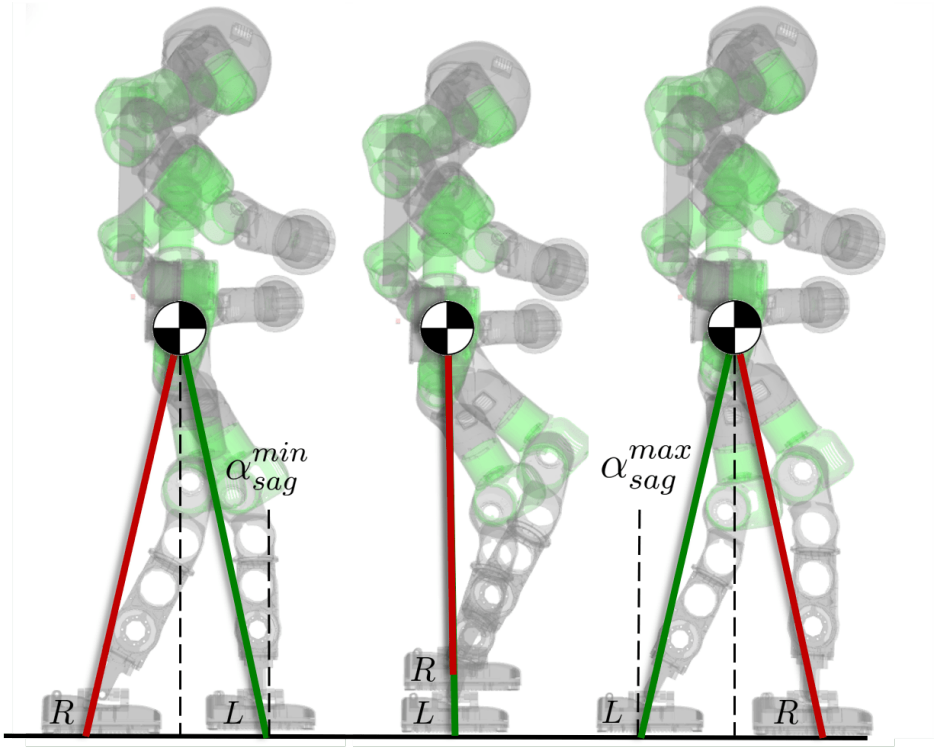


Figure 5.4 Evolution of one stepping cycle of the right leg from taking off to landing. The 2-link compass model is used as a template to impose the desired VC on the robot.

5.4 Sagittal plane: Virtual Constraints

VCs are geometric relations between state variables that bind their motion together. Setting a suitable set of VCs, the desired motion can be imposed on any n-dimensional system, by controlling a smaller set of state variables $n - n_m$, where n_m is the number of constrained variables. In the case of a biped, they coordinate the evolution of the links' position along a periodic orbit, which physically translates into a specific stepping motion.

To illustrate this, let $\boldsymbol{x} \in \mathbb{R}^n$ be the state vector of a n-link bipedal robot and $\boldsymbol{x}_c := h : \mathbb{R}^n \to \mathbb{R}^m$ a generic function that specifies the m quantities to be controlled, i.e. the relevant joints involved in locomotion, such as knees, hips or ankles. These quantities are bound together by the VCs $h(\boldsymbol{x}_c)$ to encode a specific pattern for locomotion. One VC is imposed on the mechanical system by zeroing the output of the function:

$$z = h(\boldsymbol{x}_c) - h_d(\alpha(\boldsymbol{x}_f)), \tag{5.9}$$

where h_d describes the evolution of $h(\boldsymbol{x}_c)$ as a function of $\alpha(\boldsymbol{x}_f)$, known as *phase variable*. The phase variable is defined as a function of a set of r quantities $\boldsymbol{x}_f : \mathbb{R}^n \to \mathbb{R}^r$ that remains

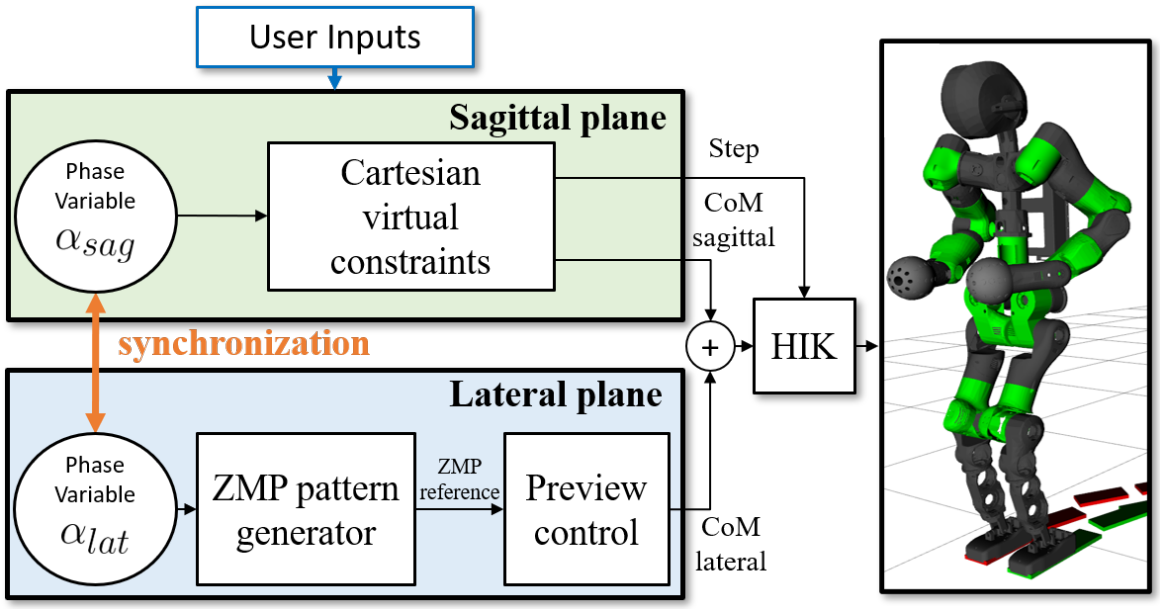


Figure 5.5 Scheme of the proposed WPG.

free. While $\alpha(\boldsymbol{x}_f)$ monotonically increases from α^{min} to α^{max}, the controlled variables $h(\boldsymbol{x}_c)$ are forced along a desired trajectory in the joint space, which corresponds to a specific stepping motion of the robot, as shown in Figure 5.4: the VCs are holonomic constraints varying the controlled variables $\boldsymbol{x}_c$ only as a function of the robot's state and not with respect of time.

Usually in literature the phasing variable $\alpha(\boldsymbol{x})$ is defined as the angle between stance leg and ground. In under-actuated robots or point feet walker models $\phi(\boldsymbol{x})$ is not controlled, and its evolution (often determined by the dynamics of a LIP) is exploited to generate a trajectory for the swing leg in such a way to enforce a periodic motion. The proposed strategy is based on the same formulation, but shifts the concept of VCs in the Cartesian space, treating them as tasks in a whole-body Stack of Task environment.

Template-space constraints

The chosen template model for a humanoid robot is the 2-link compass walker, comprising a condensed mass, which corresponds to the CoM, located at the joints connecting two massless links of equal length, as depicted in Figure 5.1. The configuration vector is $\boldsymbol{\theta} = [\theta_1, \theta_2]^\top$, where θ_1 and θ_2 are the angles describing respectively the *stance* and *swing* leg.

By taking advantage of the symmetry of the model, a simple yet effective description of the leg advancement is to force the swing leg to behave as a mirror image of the stance leg.

According to (5.9), the formulation for the compass geometry translates into the following VC:

$$y = \theta_2 - \pi + 2\theta_1 \tag{5.10}$$

where θ_1 is chosen as the phasing variable $\alpha(\boldsymbol{x})$ and θ_2 specifies the quantity to be controlled, constrained by the evolution of θ_1. Finally, z is the output to be driven to zero.

Task-space constraints

Instead of imposing a desired trajectory in the joint-space, VCs are used to coordinates Cartesian variables. A parallel formulation of the template-space VC binds the phase variable α[1] with the Cartesian quantities deemed relevant for the desired task such as the CoM or the feet position. The angle between the normal of the ground and the virtual leg connecting the CoM to the stance foot is selected as the phase variable α, as shown in Figure 5.1, which corresponds to the *tilt angle* θ_1. When the stance leg starts leaning in the sagittal direction, the angle θ_1 increases, which corresponds to a CoM displacement:

$$c = h\tan(\theta_1), \tag{5.11}$$

where h is the height of the CoM w.r.t. the ground. By formulating the step displacement in the Cartesian space of the swinging leg as a function of the configuration vector $\boldsymbol{\theta}$, it becomes:

$$s = h\tan(\theta_1) + h\tan(\pi - \theta_1 - \theta_2) \tag{5.12}$$

Substituting θ_2 in (5.10) yields the Cartesian VC, constraining s as a function of the *tilt angle* θ_1 only:

$$s = 2h\tan(\theta_1) \tag{5.13}$$

This VC links together the CoM motion and the foot position, more precisely restricting the swing leg to keep the position of the CoM in between the legs. In other words, the tilting of the robot moves the CoM which constrains the leg to follow the desired stepping motion until an impact occurs, which reset the cycle. The VC constrains the Cartesian displacement of the leg, not its actual trajectory: thus, s can be treated as a task for the whole-body IK solver. According to the model in Westervelt et al. (2007), the collisions are assumed as fully inelastic, i.e. once landed at the contact point, the swing foot does not experience slips or bounces. This assumption is imposed by setting zero velocity at the beginning and the end of the step to reduce undesired rebounds after impacts and make the walking smoother. As a

[1] From here on, the notation of $\boldsymbol{x}$ in $\alpha(\boldsymbol{x})$ is dropped for simplicity.

last note, the impact is assumed to be instantaneous, i.e. the stance leg is lifted at the same time the swing foot touches the ground: a double stance phase between the switching single stances of the legs is not considered.

Ankle actuation

By applying a suitable VC, the template model is treated as a system with the *tilt angle* θ_1 as its only DoF. Its evolution obeys the free-fall dynamics of a linear inverted pendulum (LIP), corresponding to the stance leg. Injecting the natural dynamics into the system reduces the energy consumption of the gait, but limits the flexibility of the stepping motion, which is bound to a non-controlled variable. Differently from the point feet in the template model, COMAN+ has flat soles and fully actuated ankles. This allows to directly control the *tilt angle* θ_1 of the robot and, consequently, the phase variable α. In particular, as discussed in Chapter 2, the CoM is fully controllable as long as the ZMP dwells inside the support polygon, while it becomes under-actuated if the ZMP is pushed on the edge, as the robot starts tilting with its natural dynamics. Indeed, explicitly controlling the ankles prevent to take advantage of the free-falling dynamics in the sagittal direction, which would reduce the gait's power consumption. Nonetheless, the full controllability in a system is advantageous, since it allows to regulates the behaviour of the robot, making it possible, for instance, to steer, change the feet distance in the lateral plane or increase the step stride. Additionally, controlling the CoM allows to unify the models along the two planes to guarantee a consistent system: the Cartesian constraints that link the sagittal displacement of the CoM and the trajectory of the swing leg is used to impose a constant height of the CoM along the stepping motion, so that the PC in lateral plane can rely on the LIPM. Controlling the sagittal tilt does not preclude the possibility to exploit the natural dynamics of the system: by commanding the CoM further in the forward direction, it is possible to drive the ZMP on the edge of the support polygon, introducing a "controlled fall" phase in the gait.

5.5 Lateral plane: Preview Control

The choice of relying on PC in the lateral plane is backed up by the work in Sardain and Bessonnet (2004), which shows that the dynamic component mainly resides in the sagittal plane of the foot: throughout the stepping motion, the ZMP travels in the sagittal direction until it reaches the edge of the support, while it does not move considerably along the lateral axis.

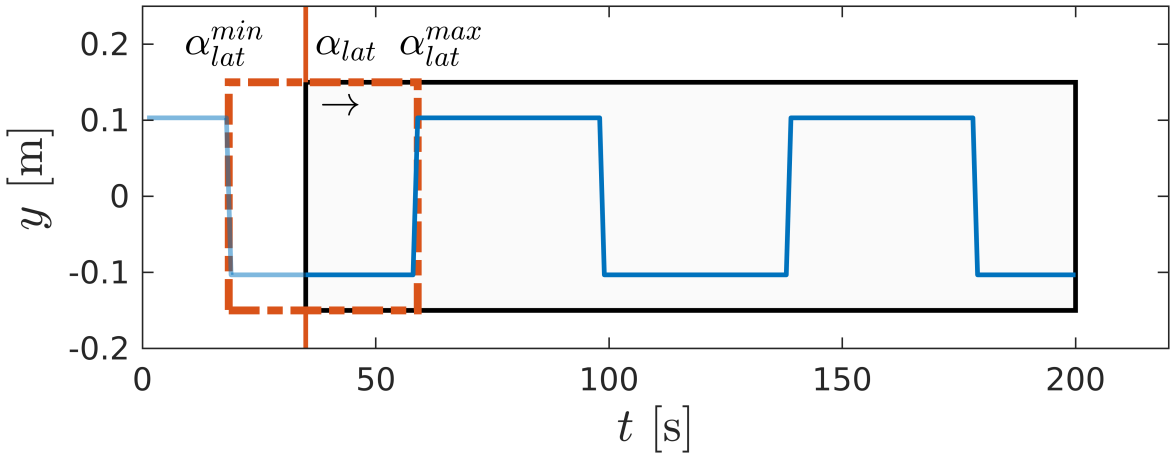


Figure 5.6 A snapshot of the preview window (in black) and the ZMP future reference (in blue) during a step. The advancement of the preview window is a function of the phase variable α_{lat} from α_{lat}^{min} to α_{lat}^{max}, and it does not depend on time.

Therefore, by modeling the robot as a LIPM and imposing a reference ZMP, the PC scheme can track it by computing a suitable CoM trajectory. Given the lateral component of center-of-mass $y_{\text{com}} \in \mathbb{R}$, the zero moment point along the lateral plane $y_{\text{zmp}} \in \mathbb{R}$ can be computed as follows:

$$y_{\text{zmp}} = y_{\text{com}} - \frac{h_{\text{com}}}{g} \ddot{y}_{\text{com}}, \tag{5.14}$$

where $h_{\text{com}} \in \mathbb{R}^+$ is the CoM height above the ground, and g is the magnitude of the gravity vector. Notice how this formulation is derived in Section 2.5.2, where the same result is obtained from the simplified model of the robot. The goal consists then in computing a lateral CoM motion $y_{\text{com}}(t)$ such that the resulting ZMP (5.14) always lies within the lateral support of some predefined footsteps.

Following the work of Kajita et al. (2003a), the ZMP equation (5.14) is interpreted as an output equation of a Linear Time-Invariant (LTI) dynamical system, in order to formulate an output tracking problem. A common choice is to select the state vector $\boldsymbol{x} \in \mathbb{R}^3$ for such a system as

$$\boldsymbol{x} = \begin{bmatrix} y_{\text{com}} & \dot{y}_{\text{com}} & \ddot{y}_{\text{com}} \end{bmatrix}^\top, \tag{5.15}$$

while the control input $\boldsymbol{u} \in \mathbb{R}$ is the lateral CoM jerk. After zero-order-hold discretization with time step T, a discrete-time LTI system of the following form is obtained:

$$\begin{cases} \boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ y_{\text{zmp},t} = C_{\text{zmp}}\boldsymbol{x}_t \end{cases}, \tag{5.16}$$

where

$$A = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, \tag{5.17}$$

and the output, according to (5.15), reads:

$$C_{\text{zmp}} = \begin{bmatrix} 1 & 0 & -\frac{h_{\text{com}}}{g} \end{bmatrix}. \tag{5.18}$$

As noted in Subsection 5.2.1, effective tracking of the ZMP output requires anticipative action, i.e. knowledge of the desired ZMP over a future horizon. Hence, the servo tracking problem is formulated within a finite-horizon LQR setting, which amounts to solving the following equality-constrained QP problem:

$$\begin{aligned} &\min_{\boldsymbol{X},\boldsymbol{U}} \sum_{k=t+1}^{t+N} Q\|C_{\text{zmp}}\boldsymbol{x}_k - y^{\text{ref}}_{\text{zmp},k}\|^2 + r\,\|\boldsymbol{u}_k\|^2 \\ &\text{s.t.} \\ &\quad \boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k, \quad k \in \{t,\ldots,t+N-1\} \end{aligned}, \tag{5.19}$$

where $y^{\text{ref}}_{\text{zmp},k}$ is the desired ZMP at the k-th future time step. The QP problem (5.19) consists in minimizing the state $\boldsymbol{X} = [\boldsymbol{x}_{t+1},\ldots,\boldsymbol{x}_{t+N}]^\top \in \mathbb{R}^{3N}$ and the input trajectory $\boldsymbol{U} = [\boldsymbol{u}_t,\ldots,\boldsymbol{u}_{t+N-1}] \in \mathbb{R}^N$ over a prediction horizon of length $N > 0$ and constrained by the dynamics of the LTI system (5.16) The weight $r > 0$ is introduced to penalize the control action.

The solution of a discrete-time finite-horizon LQR problem, if the convexity condition holds, can be found by solving the Karush-Kuhn-Tucker (KKT) conditions. For a LQR problem with equality constraints, the KKT conditions are a set of linear equations. Then, the optimization problem (5.19) can be solved from the corresponding *KKT equations*: thanks to the form of the coefficient matrix of the linear system, a highly sparse set of $n = 4N + 3N$ equations in as many unknowns, the solution can be efficiently found by exploiting the LU decomposition from the *Eigen3* library. Note that the same solution can be found via the well-known concept in control theory, the Riccati equation. The optimal control action is composed of two terms: the matrix K_{fb} is the gain of the state vector, while the matrix K_{fb} is the gain of the reference ZMP. These gains are calculated from the weights of the input r and the output Q in (5.19). Then, taking a *receding horizon* approach, at each iteration the first control input $\boldsymbol{u}_t$ is applied to the system (5.16); it can be shown that this is linearly dependent

on the current state $\boldsymbol{x}_t$, and the reference ZMP trajectory $\boldsymbol{Y}_{\text{zmp}}^{\text{ref}} \in \mathbb{R}^N$

$$\boldsymbol{u}_t = K_{\text{fb}}\boldsymbol{x}_t + K_{\text{ff}}\boldsymbol{Y}_{\text{zmp}}^{\text{ref}}, \tag{5.20}$$

where $K_{\text{fb}} \in \mathbb{R}^{1\times 3}$ is a state feedback matrix, whereas $K_{\text{ff}} \in \mathbb{R}^{1\times N}$ is a reference feed-forward matrix. Notice that the state vector $\boldsymbol{x}$ of the LIPM is not updated with the real state of the robot. Its future value is computed by integrating the current $\boldsymbol{x}_t$ over the period T, subject to the constant control action $\boldsymbol{u}_t$. Finally, the closed-loop stability of the controlled system can be assessed from the magnitude of the closed-loop eigenvalues:

$$|\lambda| < 1 \quad \forall \lambda \in \text{sp}(A + BK_{\text{fb}}) \tag{5.21}$$

If the stability criterion (5.21) is not met, then the horizon length N should be increased: consequently, more of the future ZMP reference must be fed into the controller, while guaranteeing that the resulting computational burden introduced by the additional length of the preview horizon can be handled within one control cycle. Indeed, a trade-off should be found between a N too high, which would slow down the computation without any reasonable improvement, and a N too low, which would generate undesired overshoots in the ZMP tracking.

5.6 Synchronization

While they can be divided and analyzed separately, bipedal locomotion entails a tight coupling between the motions in the sagittal and the lateral plane. In this section, a means for unifying time and phase-dependent controllers is presented to synchronize the sagittal and the planar motions of the robot. This part is focused on investigating how preview control and VCs bind together in humanoid locomotion while being structurally unrelated approaches. The preview control strategy inherently requires a future reference to track to generate the desired CoM trajectory. While the prediction window carries relevant information, the real evolution of the system may not coincide with the predicted behaviour, resulting in steps occurring before or after the expected time. The stepping motion, on the contrary, is driven by a phase variable which is not dependent on time: hence, any perturbation of the gait would cause a desynchronization of the two motions, as shown in Figure 5.7 where the independent components quickly lose the combined pattern, producing an unfeasible walking gait.

The solution proposed in this work drops the time dependency in the preview control, linking the advancement of the horizon window to the evolution of the system in the sagittal plane.

Sagittal Plane The forward displacement, i.e. the tilt of the robot measured by θ_1 in parallel with the swing foot movement, is dependent on the phase variable α_{sag} according to (5.11) and (5.13). Increasing α_{sag} from α_{sag}^{min} to α_{sag}^{max} corresponds to forcing a full stepping motion on the robot, as shown in Figure 5.4.

Lateral Plane A complete cycle in the lateral plane corresponds to a switching between the two single support phases: the reference ZMP residing within the current support polygon, i.e. the sole of the stance foot, shifts to the opposite leg at the moment of impact. This behaviour translates to the corresponding advancement of the preview window in the PC scheme. However, instead of expressing it as a function of time, its progression is bounded to the evolution of the phase variable α_{lat} from α_{lat}^{min} to α_{lat}^{max}. The WPG fills automatically

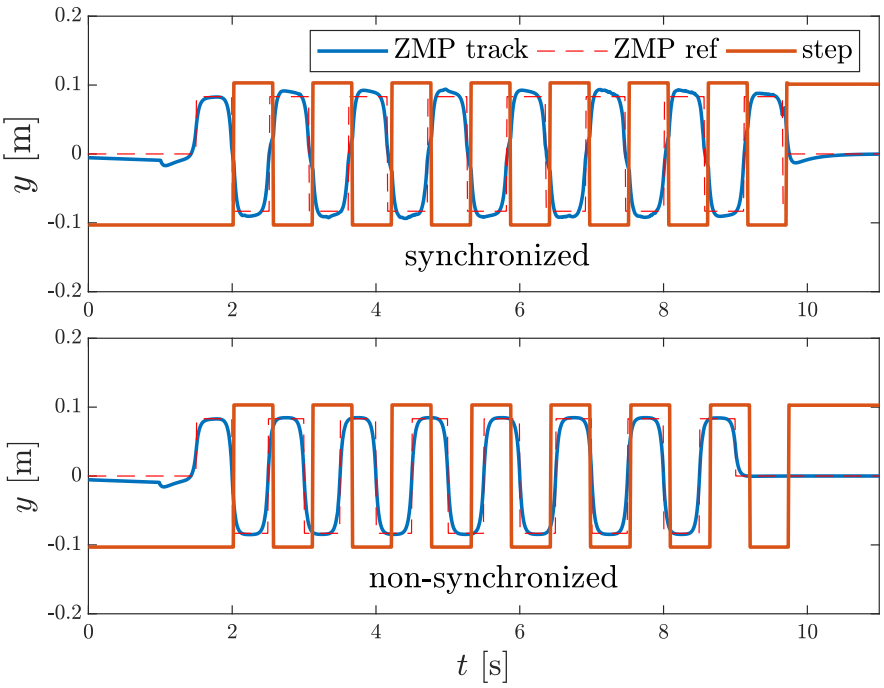


Figure 5.7 Above: synchronized behaviour of the walking motion. The stepping pattern (in orange) is always synchronized with the lateral ZMP and the corresponding CoM trajectory. Below: the lateral and sagittal planes are not synchronized, hence the stepping pattern progressively drifts.

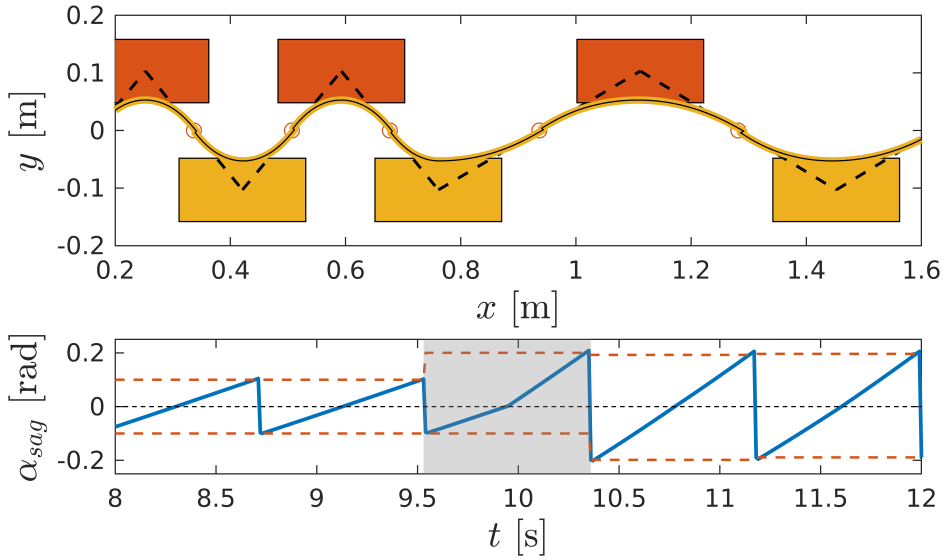


Figure 5.8 A detail of the WPG changing step stride. Top: walking pattern of the robot changing the length of the step accordingly to the new α_{sag}^{max}. Bottom: the trajectory of the phase variable α_{sag}. The grey area corresponds to the step where α_{sag}^{max} changes: in the second half of the step the steepness of α_{sag} increases, which corresponds to an increase in the CoM velocity.

the preview windows with a feasible future trajectory of the ZMP reference: as shown in Figure 5.6, its structure is divided into a first segment, corresponding to the position along the y-axis of the stance foot, a second segment, coinciding with the y-component of the goal position of the swing foot. and a final tail, consisting of a repeating pattern based on the ZMP values of the current and next steps.

Synchronization By measuring α_{sag} at each iteration of the controller and binding its evolution to α_{lat} an automatic synchronization of the two planes is obtained, resulting in a simultaneous advancement of the CoM and the swing foot in the sagittal direction and the CoM sway in the lateral plane. Specifically, α_{lat} is directly driven by α_{sag}: the monotonic increment of the sensed sagittal phase variable corresponds to the progression of the lateral one.

5.7 Implementation

The structure of the WPG is shown in Figure 5.5: the hybrid architecture generates cyclic reference trajectories, where each full cycle corresponds to a single step: the evolution of the

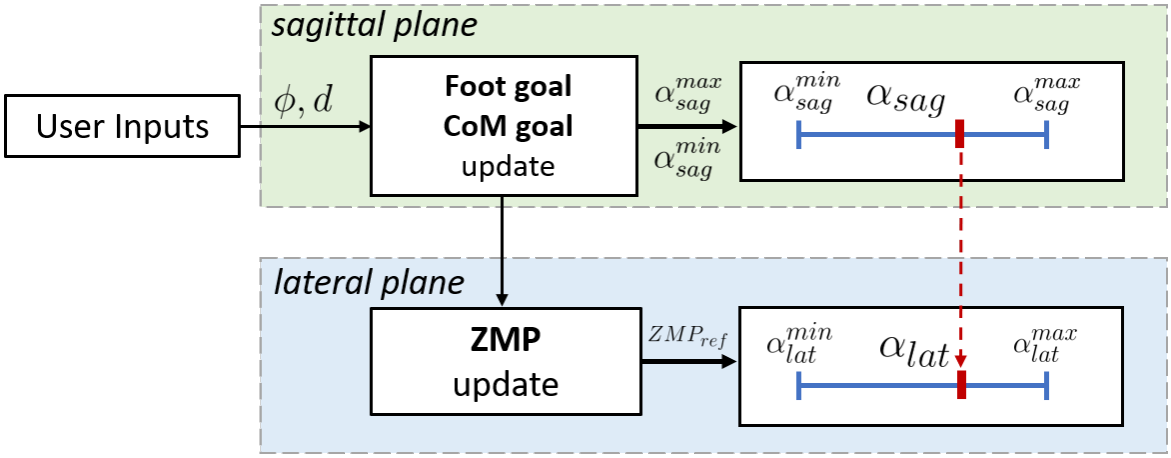


Figure 5.9 Scheme detailing the process of synchronization between lateral and sagittal planes. In the sagittal direction the foot and the CoM goals are updated by the user inputs θ and d, which modifies the extrema of the phase variable, α_{sag}^{max} and α_{sag}^{min}. In the lateral plane the ZMP reference fed to the PC is updated accordingly. The lateral phase variable α_{lat} is driven by the sagittal phase variable α_{sag}.

CoM and the swing foot on each plane is governed by the phase variables α_{sag} and α_{lat}. The evolution of the two components along the planes is synchronized to obtain a feasible gait.

5.7.1 Online Walking Pattern Generator

The step can be modulated by a few user-defined dynamic parameters that can be modified online:

- α_{sag}^{max}: the maximum inclination, i.e. stride of the step;
- ϕ: the steering angle, i.e. to the angle between the initial and the current direction of the CoM;
- d: the lateral distance between the feet.

the phase variable α_{sag}: α_{sag}^{max}, according to (5.11), directly controls the CoM. The *tilt* of the robot θ_1 is continuously sensed, and governs the evolution of step: in fact, the displacement of the CoM in one cycle constrains the length of the step through the VC (5.13): therefore, any desired step stride can be imposed by modifying the interval $\left[\alpha_{sag}^{min}, \alpha_{sag}^{max}\right]$. At each cycle, α_{sag}^{min} is registered as the sensed θ_1, flipped along the central point $\alpha_{sag} = 0$, where the angle between the two legs is zero, as depicted in Figure 5.4. Its value cannot be modified, as it depends on the past behaviour of the robot. Differently, α_{sag}^{max} can be issued by

the user to modify the interval over which the phase variable acts. By doing so, the CoM displacements can be adjusted and its velocity during the step changes accordingly, as shown in Figure 5.8.

At each impact the WPG switches the swing leg *sw* with the stance leg *st* and vice-versa. Additionally, the parameters in the WPG are changed according to the upcoming step:

- α_{sag}^{min}: updated with the sensed θ_1;
- α_{sag}^{max}: in the absence of user inputs, kept as the last maximum registered inclination θ_1;
- ϕ: if no commands were issued by the user, kept as the last heading angle;
- s, c: the swing foot (5.12) and CoM (5.11) displacement, computed as a function of α_{sag}^{min} and α_{sag}^{max} and the corresponding swing foot goal position;

As noted before, the CoM lies in the middle of the two feet in the sagittal direction and when $\alpha_{sag} = 0$ the CoM is above the stance foot (5.4). Given the updated α_{sag}^{max} and ϕ, the next footstep is automatically generated by the WPG during the preceding step. Moreover, at each control cycle, the receding window is updated with a suitable ZMP reference to accommodate the gait, as shown in Figure 5.9. The proposed WPG exploits a simple state machine to switch between the walking phases, such as starting, stopping, and receiving user-inputs, which are issued through a simple communication interface. In particular, it runs until a stop command is issued, it accepts on-line commands such as step length, feet distance d and desired heading angle ϕ.

5.7.2 Whole-Body Inverse-Kinematics

The integrated control was enclosed in a state machine to manage the walking phases, i.e. starting, stopping and walking.

The Cartesian trajectories computed by the walking pattern generator are realized in the robot thanks to the whole-body inverse kinematics framework CartesI/O (Laurenzi et al., 2019). It allows to specify Cartesian tasks for the robot which are carried out in a stack of task (SoT) fashion: lower priority task run in the null space of higher priority ones, so as to guarantee the execution of critical tasks without any low-priority one interfering. The

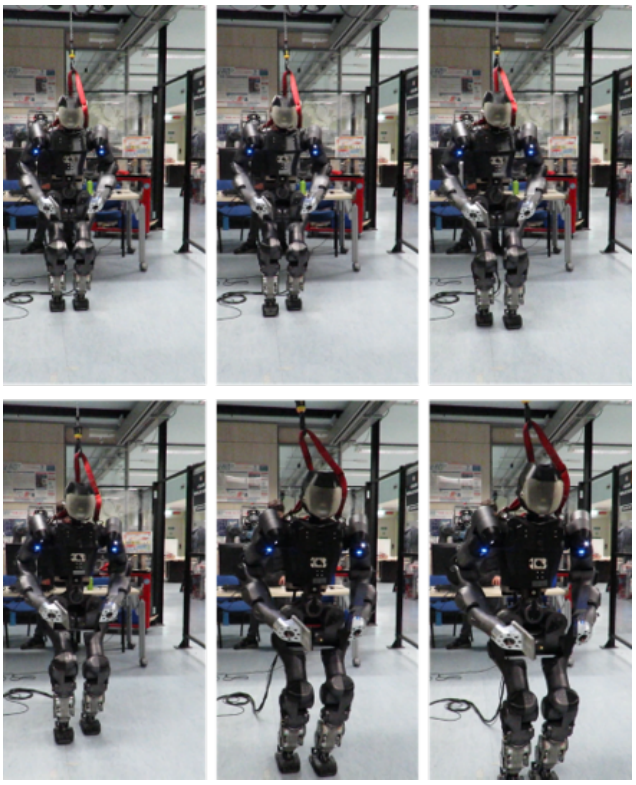


Figure 5.10 COMAN+ walking 15 steps, with 0.15 m step length, with the proposed WPG.

following SoT was designed for the WPG:

$$\left(\begin{array}{c}\left({}^{\mathrm{W}}\mathcal{T}_{\mathrm{LFoot}}+{}^{\mathrm{W}}\mathcal{T}_{\mathrm{RFoot}}+\mathcal{T}_{\substack{\mathrm{Posture}\\\mathrm{TorsoRoll}}}\right)/\\ {}^{\mathrm{W}}\mathcal{T}_{\mathrm{CoM}}/\\ {}^{\mathrm{W}}\mathcal{T}^{RPY}_{\mathrm{Waist}}/\\ \mathcal{T}_{\mathrm{Posture}}\end{array}\right) << \left(\mathcal{C}_{\substack{\mathrm{Joint}\\\mathrm{Lims}}}+\mathcal{C}_{\substack{\mathrm{Vel.}\\\mathrm{Lims}}}\right) \tag{5.22}$$

The / symbol is used to impose *strict* hierarchy among sets of tasks which are in a relation of *soft* hierarchy (+ symbol). The << symbol takes into account constraints in the task execution such as joint limits and joint velocity limits. The description ${}^{\mathrm{A}}\mathcal{T}_B$ define a task of the frame B expressed w.r.t. the frame A. If A is not specified, task $\mathcal{T}_B$ is expressed in joint space. In the chosen stack, ${}^{\mathrm{W}}\mathcal{T}_B$ refers to a generic task B w.r.t. the world frame. The stack in (5.22) is then solved by a cascade of QPs. The task on the Torso $\mathcal{T}_{\substack{\mathrm{Posture}\\\mathrm{TorsoRoll}}}$ minimizes the lateral swing of the robot at each step by keeping it perpendicular to the ground. The tasks on the feet ${}^{\mathrm{W}}\mathcal{T}_{\mathrm{LFoot}}$ and ${}^{\mathrm{W}}\mathcal{T}_{\mathrm{RFoot}}$ corresponds to the contact tasks, therefore are assigned the highest priority. All the other tasks, including the CoM ${}^{\mathrm{W}}\mathcal{T}_{\mathrm{CoM}}$, act in the null-space of the

contacts to guarantee a consistent solution within the base under-actuation. The hierarchy of the tasks is not unique and can be rearranged, but empirical results showed the viability of the chosen SoT. The Postural task guarantees that all the other joints not involved in the walking motion are maintained as close as possible to a given neutral configuration.

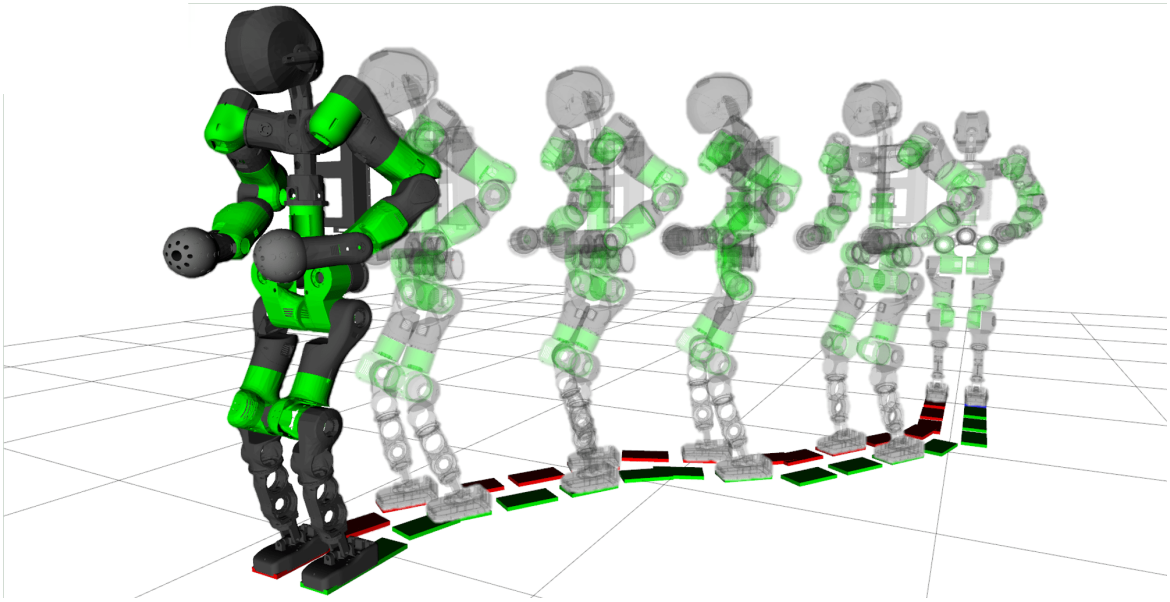


Figure 5.11 Sequence of frames of the COMAN+ curving two times left, for a total steering angle of 60 degrees, and steering back in the straight direction.

5.8 Omni-directional Walking

Due to the different nature of the lateral and sagittal plane, the hybrid WPG is not trivially generalized for omni-directional walking. This section describes how the WPG is enhanced to steer in any given direction, as shown in Figure 5.11.

The *tilt angle* θ_1 of a three-dimensional robot is defined as:

$$\theta_1 = \arctan\left(\frac{{}^{com}x_{st}}{{}^{com}z_{st}}\right), \tag{5.23}$$

where ${}^{com}x_{st}$ and ${}^{com}z_{st}$ are the x and z components of the *stance foot* ankle position ${}^{com}\boldsymbol{p}_{st} \in \mathbb{R}^3$ w.r.t. a reference frame centered at the CoM and oriented with the x-axis aligned to the direction of motion, as clearly depicted in Figure 5.12. Whenever a new angle ϕ is commanded, the reference frame at the CoM is rotated by ϕ along the z-axis. The WPG computes the corresponding steering step to change the direction of motion from the heading

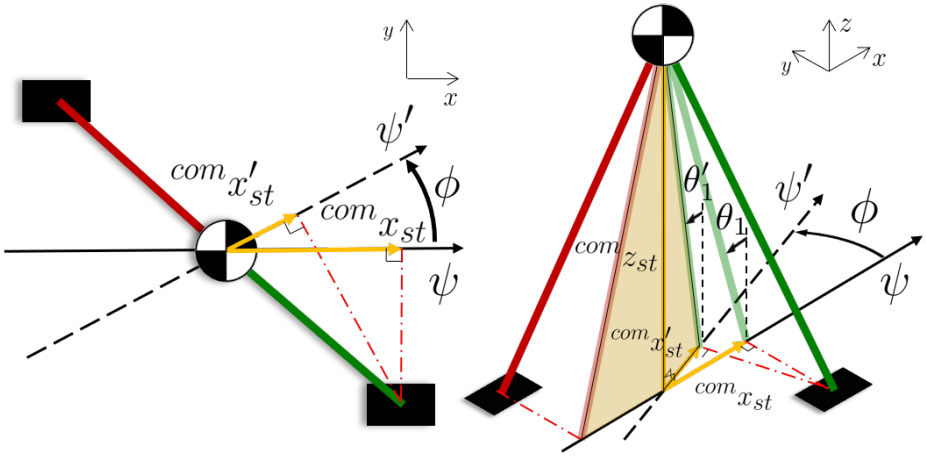


Figure 5.12 Compass model (left: xy-plane view, right: 3D space representation) during a steering step from heading ψ to ψ'.

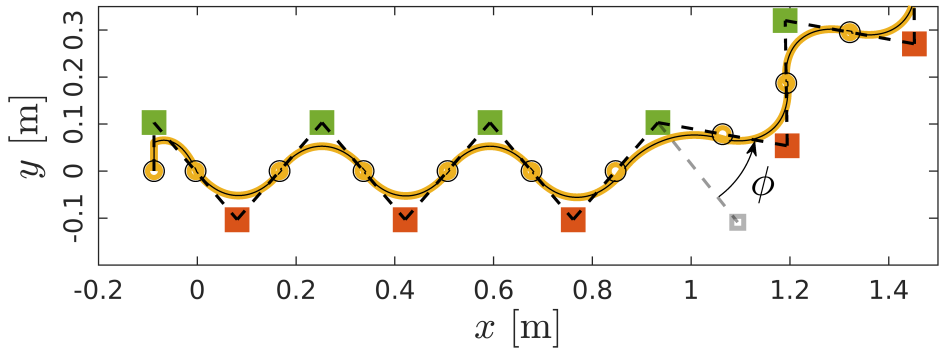
angle ψ to the new one $\psi' = \psi + \phi$. At this point, the stance ankle's position is recomputed according to:

$$^{com}\boldsymbol{p}'_{st} = \boldsymbol{R}(\phi)^{com}\boldsymbol{p}_{st}, \tag{5.24}$$

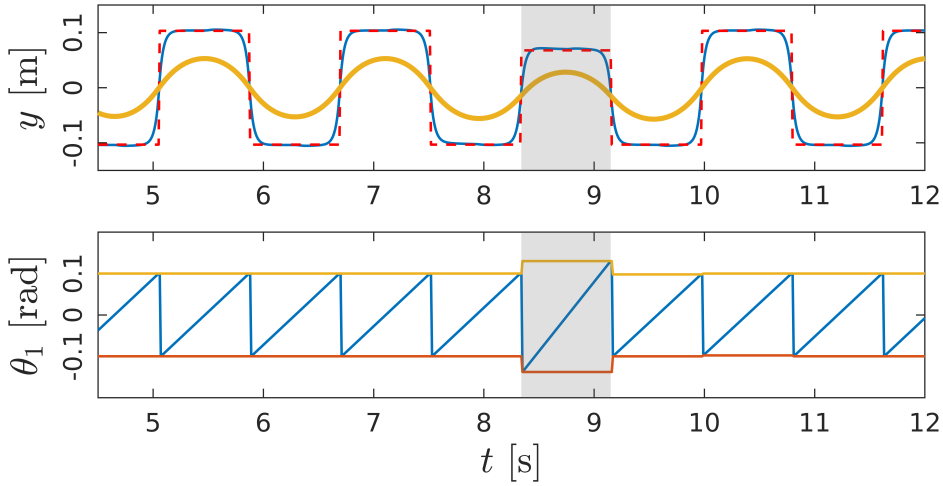
where $\boldsymbol{R}(\phi) \in \mathbb{R}^{3\times3}$ is the rotation matrix representing a rotation by ϕ on the horizontal plane. In Figure 5.12 is shown the rotation of the sagittal plane rotates by ϕ: the new projection of the ankle position modifies the distance between the stance ankle and the CoM, which corresponds to a change in the *tilt angle* θ'_1 of the robot: therefore, once the rotation (5.24) is applied, at the beginning of the steering step a new θ'_1 is computed from (5.23). The steering motion is designed to maintain at a constant length d the legs in the lateral direction after the change of heading. The remaining parameters are determined through the following procedure:

- compute $^{st}\boldsymbol{T}^{goal}_{sw}$ (goal pose of the *swing foot* w.r.t the *stance foot*) based on the current heading ψ and the phase variable α^{max};
- compute $\boldsymbol{p}^{goal}_{com}$ (the goal position of the CoM) as the middle point between the *stance foot* and the new *swing foot* position;
- rotate $^{st}\boldsymbol{T}^{goal}_{sw}$ of ϕ along the z-axis.
- set α^{max} according to (5.11), so that the CoM position is $\boldsymbol{p}^{goal}_{com}$ at the end of the step;
- update α^{min} with the new *tilt angle* θ'_1 computed using (5.23) and (5.24).

The evolution of α from its minimum until it reaches α^{max} governs the advancement of the CoM and the swing leg until the foot impacts the ground. At the impact, the previous heading



(a) The pose of the left (in green) and right (in red) sole and the CoM trajectory (in yellow).



(b) Top: ZMP reference for the PC. Bottom: the evolution of the tilt angle θ_1. The grey area highlights the steering step.

Figure 5.13 Output trajectories of the WPG when the robot changes heading direction, steering of ϕ.

angle ψ is updated with the steering angle ϕ issued by the user, and the robot advance along the new direction ϕ'. The lateral distance between the right and the left sole d is kept constant. Similarly, the step stride determined by the maximum forward tilt of the robot α^{max} remains constant before and after the curve. However, during the steering motion, both the parameters d and α^{max} are changed due to the rotation of the CoM reference frame to accommodate the desired step. Consequently, the ZMP trajectory fed to the PC and the phase variable are updated accordingly, as shown in Figure 5.13.

5.9 Experiments

While being fully torque-controlled thanks to torque sensors present at each joint, in the following experiments a position control has been employed. The WPG runs at 100Hz in a ROS node and exposes a simple ROS interface for issuing commands to the robot. The walking pattern generator is validated through two separate walking experiments, performed with two different step strides: the first one consists of a 5 step walk with 0.2 m stride length, while the second is a longer walk of 15 steps with 0.15 m stride length. The selected step clearance is 0.05 m, and the step duration of 0.5 s for both experiments. According to the scheme in Figure 5.5, the references (CoM and step trajectories) sent to the robot are generated independently in the two planes and then synchronized. In particular, the graph in Figure 5.14 depicts the commanded step and CoM trajectory in the sagittal plane: the phasing variable θ_{sag} and θ_{lat} constrains their evolution, resetting at each cycle after an impact occurs.

Similarly, the graph in Figure 5.15 shows the evolution in time of the ZMP and CoM in the lateral plane. The reference generated by the ZMP pattern generator is tracked by the preview controller, which outputs the lateral CoM reference commanded to the robot.

A ZMP pattern is computed from the Force/Torque sensors readings in the robot's soles. The graph in Figure 5.20 shows how the ZMP is well inside the support polygon of COMAN+ (represented as gray shaded). Discrepancies between the real ZMP and the commanded ZMP are mainly due to the elasticity of the structure and the inherent passive compliance of the robot, which tends to bend during single-stance phases. A sequence of frames of COMAN+ walking during the two experiments are shown in Figure 5.10 and Figure 5.21, taken from the relative videos of the walking[2].

To assess the validity of the steering algorithm, a set of dedicated experiments was carried out in simulation and on the real robot. Here one meaningful example in simulation is detailed. COMAN+ walks 32 steps, with a 0.05 m step clearance, a 0.4 s step duration and a 0.28 m step stride. The heading changes of $\phi = -30^\circ$ at step 5 and 13 and of $\phi = 30^\circ$ at step 19 and 27. Figure 5.16 shows the projection of the walking pattern on the x-y plane, while 5.19 depicts the progression of the soles in space. Notice how, compared to the previous experiments, a higher gait velocity corresponds to smaller lateral swings of the CoM.

Four steering angles are issued to the robot in total: $\phi = -30^\circ$ at step 5 and 13 and $\phi = 30^\circ$ at step 19 and 27. Due to the hybrid nature of WPG, the *maximum steering angle* in a single step is 30 deg: commanding a higher value would cause the robot to lose balance. When the steering motion is initiated, the PC is only responsible for the CoM trajectory in

[2]https://www.youtube.com/playlist?list=PL7c1ZKncPan77Fl36GEesprm13BgSsTxf

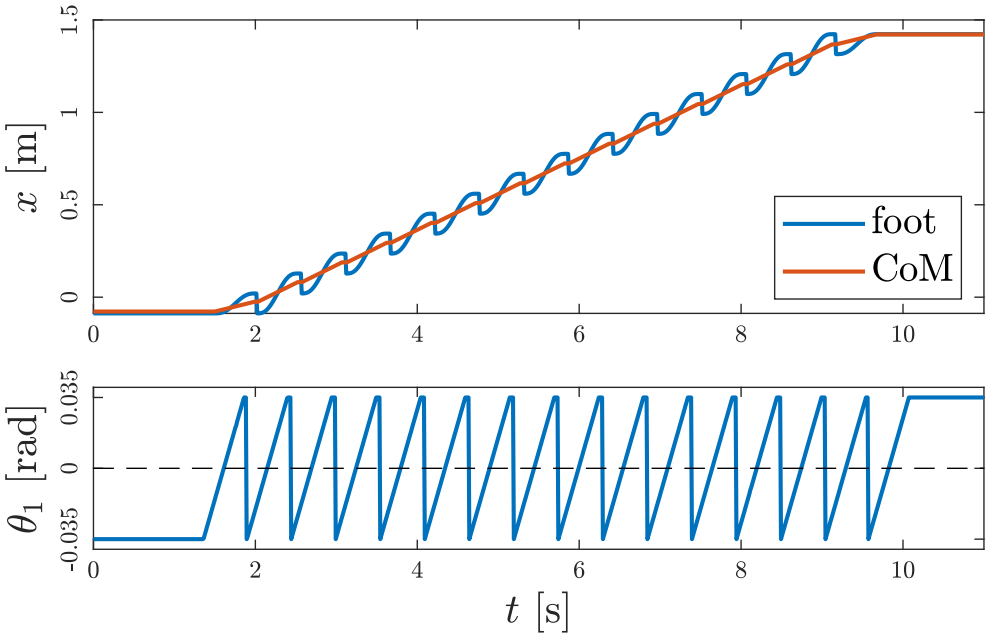


Figure 5.14 Sagittal plane: the CoM and step trajectory evolution as a function of the phasing variable θ_1, corresponding to the robot tilt. For each cycle, the phasing variable ranges from θ_1^{min} to θ_1^{max}.

the previous lateral plane, and a component of the CoM trajectory in the new direction is not guaranteed to belong to the support polygon. The *maximum step length* change is 0.4m, from a step of 0.1m to a step of 0.5m, while the *maximum speed* for stable walking in simulation is 1.25 m/s. Similarly to Figure 5.15, Figure 5.17 the evolution of the ZMP reference w.r.t. time and the phase variable θ_{sag} along the sagittal direction. A walking experiment in the simulation environment Gazebo is illustrated as a sequence of frames in Figure 5.18. To validate the effectiveness of the WPG on real hardware, it is deployed COMAN+: the robot walks for 10 steps with 0.23 m stride length, step clearance of 0.05 m and step duration of 0.4 s. A steering angle of 10° was issued at step 3. While the experiment is successful, higher steering angles are not feasible on the robot: in fact, the WPG alone does not guarantee stable walking. Without a stabilization layer between the robot and the open-loop trajectories commanded by the WPG, COMAN+ cannot react against unwanted losses of balance or reject external disturbances. A collection of videos of the experiments is provided, both in simulation and on the real robot[3].

[3]https://www.youtube.com/playlist?list=PL7c1ZKncPan7yphDxvZDtaqzmgCRtpT5-

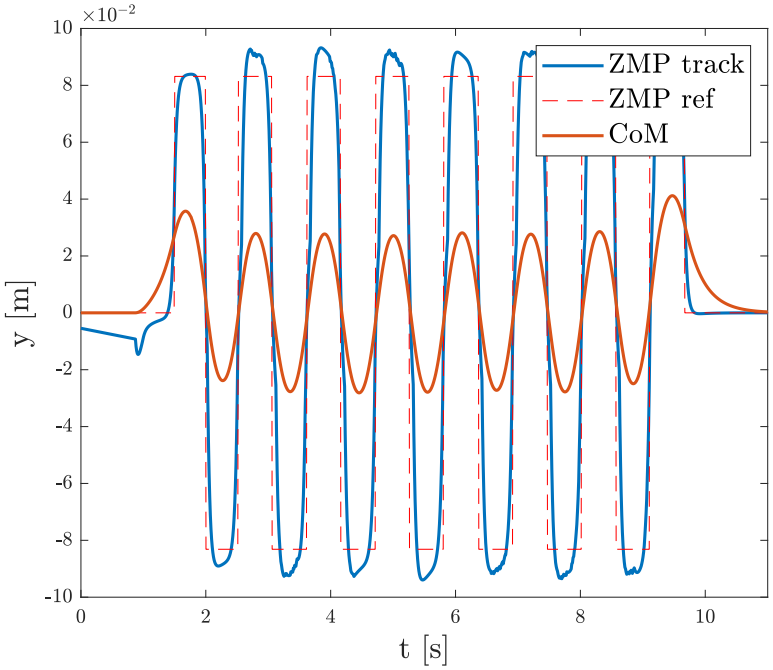


Figure 5.15 Lateral plane: the ZMP input reference for the preview controller (in red) and the tracked ZMP (in blue). The output of the preview controller is the CoM (in orange), commanded to the robot.

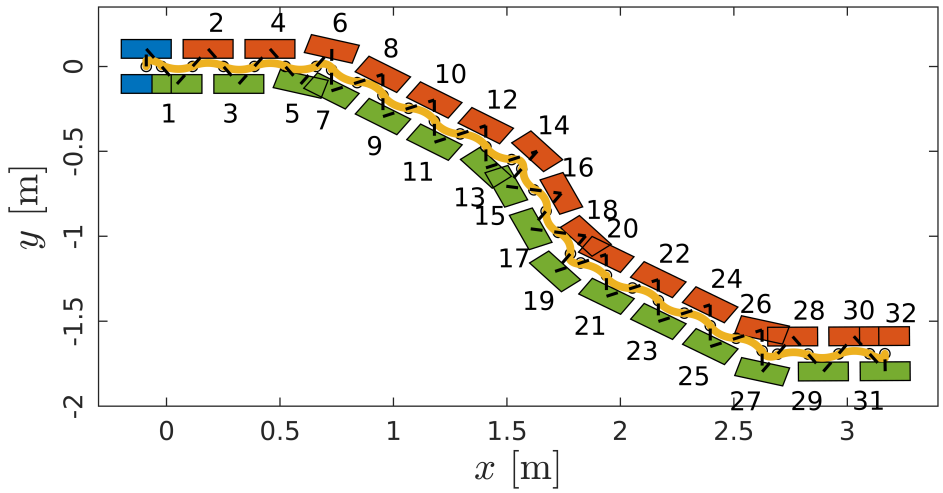


Figure 5.16 Walking pattern consisting of the CoM trajectory (in yellow) and the footsteps (initial pose in blue, left and right soles in red and green respectively).

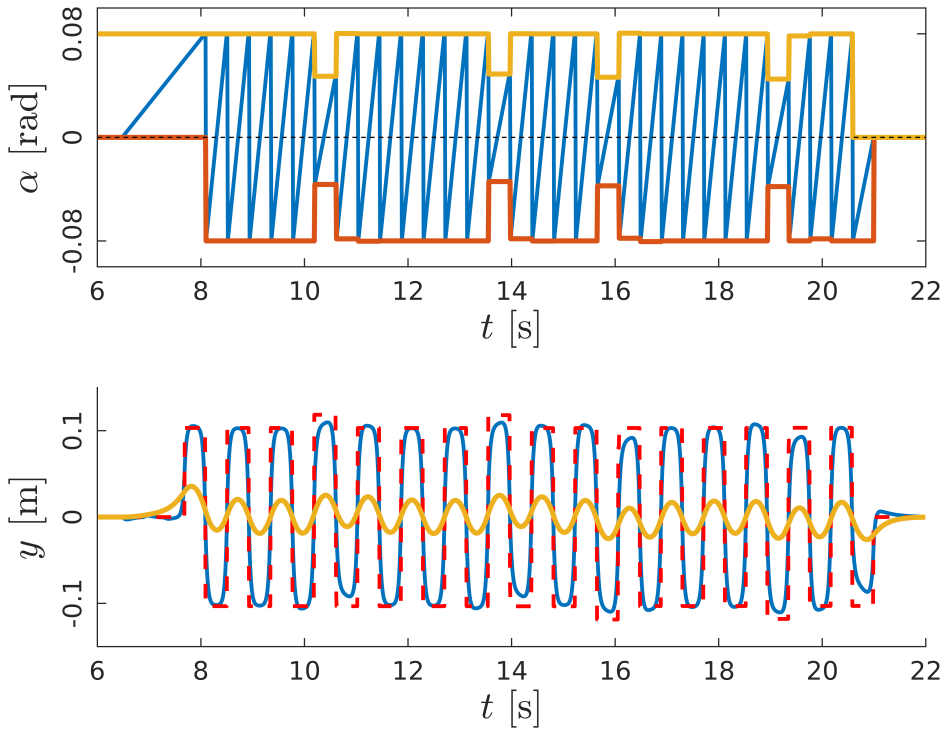


Figure 5.17 Top: evolution of the phase variable α_{sag} (in blue) in the sagittal plane during a walk, increasing at each cycle from α_{sag}^{min} (in red) to α_{sag}^{max} (in yellow). Bottom: the ZMP input reference of the PC (in red) and the tracked ZMP (in blue). The output of the PC is the CoM trajectory (in yellow) commanded to the robot.

5.10 Discussion

5.10.1 Hybrid approach

Virtual constraints enforce simple relations between state variables: they are a set of constraints imposed with control inputs that effectively simplify a complex system such as a humanoid robot into a lower-dimensional system. Imposing virtual constraints reduces the degrees of freedom of a robot, defining a set of controlled variables and a gait phasing variable to which the controlled variables depend. The somehow burdensome trajectory optimization in the classical implementation of VCs is dropped, which entails joint-trajectory optimization to find energy-efficient gaits. Usually, in literature one desired gait is obtained through an offline optimization of the robot parameters. Each optimized orbit, or a family of similar orbits, is collected in a library to allow the robot to select the walking gait among the constructed ones. Indeed, this limits the possibility to tune a walking pattern on the fly, leaving less margin for versatility. Furthermore, VCs usually binds joint variables together, making it a strategy heavily dependent on the kinematics of the robot. Instead, in the pre-

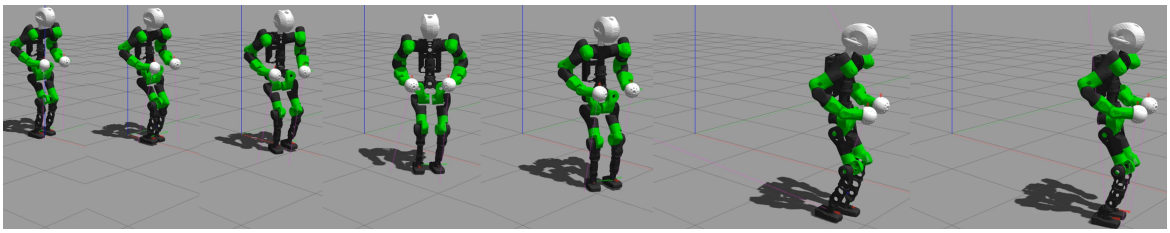


Figure 5.18 COMAN+ walking 32 steps in a GAZEBO simulation while changing the heading direction.

sented method, VCs are encoded in the Cartesian space, imposing hierarchies between tasks and designing template trajectories that can be tuned and realized on line. This allows a light and flexible tool for stepping: step length, step time, and step clearing can be easily defined on the fly. Furthermore, joint space optimization requires proof that the stepping is realistic: here, similarly to the method in Kajita et al. (2017), kinematics poses corresponding to desired gaits are designed in the sagittal plane, while the preview controller ensures a stable trajectory of the CoM in the lateral plane. Note how the design of the stepping motion is not limited by the ZMP criterion: in the sagittal plane the ZMP can move freely, since it is guaranteed to lay inside the support polygon only in the lateral direction. In doing so, a more dynamical strategy is exploited without discarding the well-known LIPM theory, effectively combining the two methods. Merging the two strategies brought interesting results: the walking pattern generator is lighter in terms of computational complexity than the classical ones and it results in open-loop stable walking of the real robot when no external disturbances are applied.

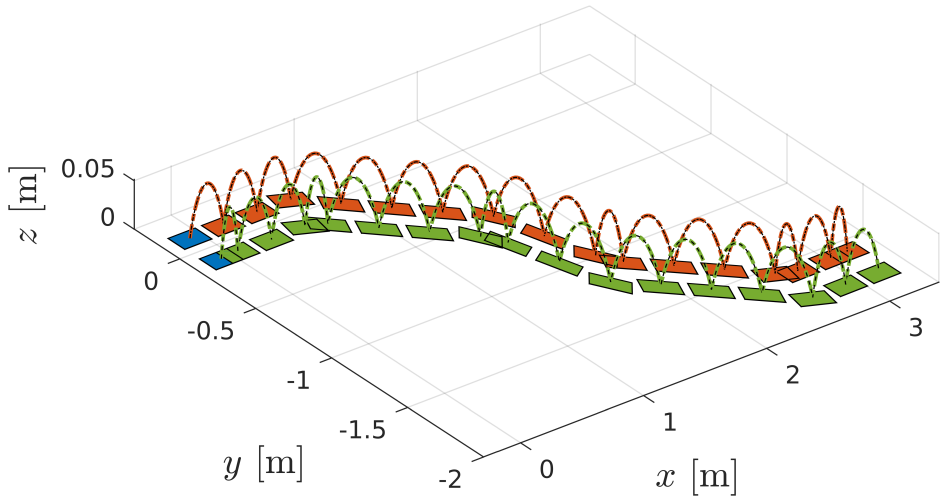


Figure 5.19 Detail of the swinging foot trajectory with a 0.05 m step clearance.

On the other hand, the energy consumption of the proposed WPG is higher compared to the classical HZD approach, where a portion of the stepping motion is entirely governed by the falling dynamic of the system. In fact, limit cycle approaches focus on generating efficient gaits: they allow zero feedback gains or, in general, lower feedback gains than ZMP approaches. High feedback gains results in a higher energy expenditure because they actively fights the natural dynamics of the system to maintain the reference motion. Oppositely, HZD gaits takes advantage of the natural dynamics to describe the desired gait. Instead of exploiting the natural dynamic of the inverted pendulum, the CoM trajectory of the hybrid WPG is fully controlled, similarly to pendulum-based approaches. The major downside of this choice resides in limiting the dynamic component of the motion. However, the "controlled" fall can be performed only by closing the loop on the robot: a state estimator reading the tilt angle while the robot is moving would allow to exploit the free-fall dynamics of the stance leg. Differently, the current formulation entails a trajectory commanded to the CoM, which is selected to be linear. However, the CoM evolution could be optimized to minimize a desired cost function, such as power consumption. Notice how modifying the CoM trajectory amounts to change the behaviour of the robot during the gait, due to the direct relationship between the CoM position and the *tilt* of the robot. Another inherent weakness of the WPG is the limited extent of modification of the gait in one single step: since the commanded gait is a cyclic motion, the robot does not sustain experience abrupt changes in the direction of movement or the step length, which would destabilize the robot. The contribution of this work was mainly focused on the development of the hybrid WPG. To integrate it on COMAN+ as a stable feature, future works will target the implementation issues, the most relevant being the lack of a stabilization layer: closing the loop on the actual robot would allow a more reactive framework capable of rejecting disturbances and sustain external pushes.

5.10.2 Results and future directions

This work presented a novel hybrid walking: the algorithm is characterized by a low computational complexity, and it can be extended for a more robust, dynamic walking. In future work, the implementation of a stabilizer on top of the walking pattern generator will be targeted, exploiting whole-body motions to enhance the robustness of the walking. Finally, recovery strategies against external disturbances will be added. As soon as the WPG is reliably deployed on the robot, a comparative study will be performed to to assess the cost of transport using the hybrid WPG and a classical ZMP approach.

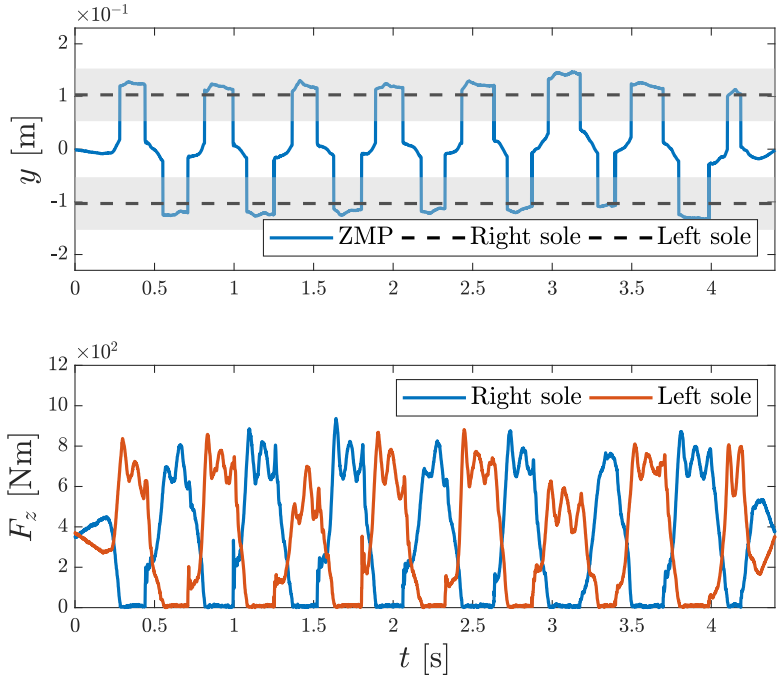


Figure 5.20 Top: the ZMP computed using the force and torque readings. The grey areas represents respectively the right sole and left sole supports. Bottom: the force readings from the sensors in the right sole and the left sole.

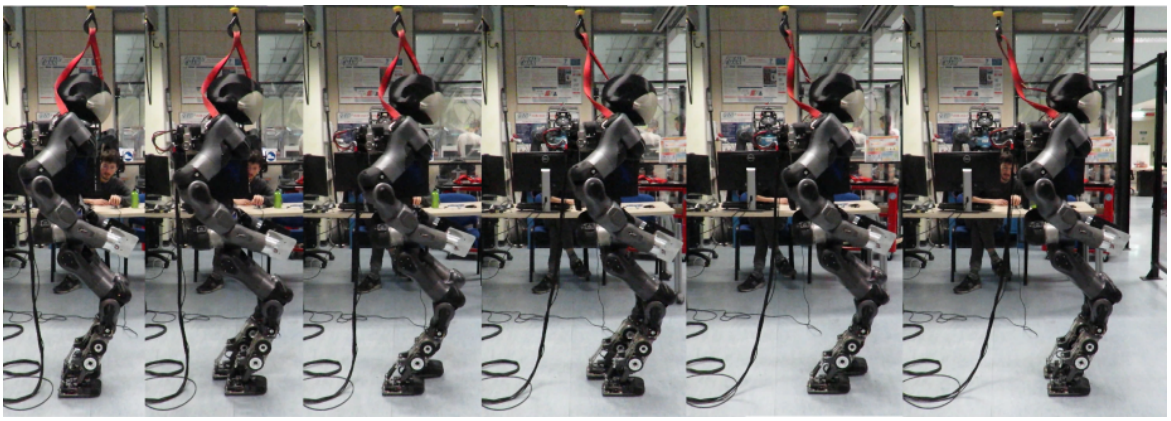


Figure 5.21 COMAN+ walking forward 5 steps, with 0.2 m step length, using the proposed walking pattern generator.

The proposed WPG generates whole-body trajectories producing feasible omni-directional walking gaits based on two structurally different methods. Towards a dynamic motion, it implements VCs on the sagittal plane, while using a ZMP-based technique similar to the well-established PC for lateral control. The proposed strategy allows to perform omni-directional walking without any burdensome parameters tuning: the stepping motion is generated accord-

ing to a few user inputs such as duration, length and stride of the desired step. Furthermore, its low computational complexity allows a fast deployment on real hardware. Simulation experiments validate the effectiveness of the strategy: the robot can walk forward, backward, steer and change walking parameters according to user-inputs. Experiments on COMAN+ show promising results: the robot can walk straight without losing balance and change its heading direction, as long as the issued steering angle is not too high. In fact, without a stabilization controller, the robot is vulnerable to model inaccuracies and external disturbances. Future works will focus on the robust deployment of this strategy on COMAN+, closing the loop on the real system and endowing the robot with stabilization techniques, such as active compliance, step recovery or whole-body adaptation.

Chapter 6

Balancing Strategies

The last chapter introduced a hybrid WPG, which merged two structurally different strategies, virtual constraints and preview control, to generate feasible references for walking. The approach was adopted following some observations in bio-mechanics, showing how the dynamic motion of the human walk is mainly generated by the sagittal component of the stepping. Tests ran in the simulation environment showed promising results, endorsed by successful experiments on COMAN+. Ideally, in the absence of real-world inaccuracies, discrepancies between the real robot and its model, tracking errors, external forces, etc., the WPG alone would be enough for locomotion. Unfortunately, this is not the case: open-loop trajectories sent to the robot without a stabilization layer are ultimately destined to diverge from a balanced gait. Two general approaches can be found for stabilization and disturbance rejections:

- embedding the stabilization layer into the WPG, closing the loop between reference sent to the robot and data sensed from the real hardware;
- adding a stabilization layer on the WPG, prompting correcting actions based on the sensed state of the robot.

This chapter presents stabilization techniques developed for the humanoid COMAN+, taking into account the above mentioned intrinsic flaws. The chosen approach is building a stabilizer on top of the WPG, effectively decoupling the planning and the control of the humanoid platform. The choice fell on this option to have a separate stabilization module that could be deployed independently of the task assigned to the robot. Indeed, these balancing behaviours are conceived for enhancing the robustness of the walking and withstand moderate disturbances. Big thrusts, highly cluttered environments and rough terrains require to be addressed specifically, as the controller should integrate several recovery strategies exploiting

whole-body motion and stepping adaptation. Two families of stabilizers were developed for the robot COMAN+ to compare different approaches:

- compliant stabilizer (Zhou et al., 2014), adjusts the CoM of the robot in response to changes in the CoP;
- ankle and torso stabilizer, command corrections at the ankle and torso level, in response to changes in the CP, to generate lateral and sagittal motions to reject external forces.

6.1 State of the art

Passive and active compliance are established methods to reject minor disturbances and small perturbations, reduce model inaccuracies, mitigate impacts with the environment and, generally, enhancing the robustness of the robot self-balance. Considering the entire viability kernel introduced in 2.5.5, these strategies belong to the region of "local stabilization", i.e. any viable behaviour of the robot given the current contacts with the environment. Indeed, passive compliance fits in: being structural, it does not require any control action, but has been proven to be beneficial for humanoids robots, both for locomotion and interaction (Pratt and Williamson, 1995; Tsagarakis et al., 2017; Tsagarakis et al., 2013). Active compliance has been implemented using different structures, targeting different cases: joint force sensing and compliant control were first used for a humanoid in Hyon et al. (2007). An admittance controller, in synergy with the intrinsic elasticity of the robot, was devised to dissipates unwanted elastic energy (Li et al., 2012a,b). The LIPM was enhanced with a spring-damper to embed active compliance in the system (Zhou et al., 2014), and impact absorption was achieved though nonlinear compliant control (Hashimoto et al., 2006).

Strategies that involved active reactions were mainly inspired by humans and derived by the centroidal dynamics of the robot: successful results were obtained in Kajita et al. (2003b) and Lee and Goswami (2010) by explicitly including the compensation of angular momentum to the stabilization controller, in addition to the linear one. The work in Ott et al. (2011) used contact force optimization to redistribute unwanted wrenches on the contacts, and a similar approach was suggested in Laurenzi et al. (2018), by post-optimizing the contact torques. Other techniques involved modifying the references sent to the robot in response to external disturbances: reactive stepping was introduced in Morisawa et al. (2009) and a similar strategy was proposed in Urata et al. (2011), which employed fast trajectory generation to recompute the step placement. Some works exploited an MPC formulation to generate step adjustments (Diedam et al., 2008; Griffin and Leonessa, 2016; Herdt et al.,

2010a), while step timing adjustments were included in the loop by the work in Griffin et al. (2017). Finally, some methods resort to discrete adjustments, such as the chest posture control in Kajita et al. (2010), the integral control in Stephens (2007) or recovery stepping (Stephens and Atkeson, 2010). All of the strategies are based on the relevant indices of humanoid self-balance such as the ZMP, the CP and the centroidal dynamics (see Chapter 2, and inspired by human behaviours.

6.2 Compliant Stabilizer

Humans preserve their equilibrium using many different strategies, the nature of which mainly depends on the features of the disturbance, i.e. magnitude, direction, duration, the current body configuration and the surrounding environments posing constraints or supports. In general, two categories can be found:

- *passive reactions*: body adaptations to dissipate the undesired energy injected by external disturbances;
- *active recovery*: deliberate motions performed by the human to keep equilibrium.

The first strategy assumes that small disturbances can be rejected by passive reactions: in particular for a robotic system, any motion that absorbs the extra energy injected by an external perturbation and stabilize the body. This approach includes a plethora of different actions which ultimately corresponds to changing the linear and the angular momentum at the CoM. Any controller belonging to this category is denoted in Zhou et al. (2014) as "*local stabilizer*". The second strategy becomes necessary if the magnitude of the disturbance raises over a threshold: as discussed in Section 2.4.4, the robot can only exert forces via the contacts with the environment. For any given contact configuration of the robot, it means that the viable corrective actions are limited by the support polygon. Bigger perturbation can only be counterbalanced by modifying the support polygon, i.e. taking a step or leaning on a surface with a limb. These controllers are denoted as "*global stabilizer*", as they belong to a larger attraction region of the viability kernel.

The strategy proposed in Zhou et al. (2014) enforces stability utilizing a *local stabilizer*. Based on force/torque readings at the soles, it generates correcting references for the CoM of the robot. This relationship can be classified as "active compliance", as it indirectly controls the reaction force by commanding corrections in the CoM reference position. The controller augment the cart-table model with a spring-damper system. In particular, the dynamics of

the model are:

$$\tau_{ext} = z_c \left(m\ddot{x} + K_s \left(x_d - x \right) \right) + mgx \tag{6.1}$$

where K_s is the natural spring coefficient of the system and τ_{ext} represents the resultant of the external forces applied on the robot. The spring-damper effect can be injected in the problem by setting the external torques so as to obtain the desired dynamics:

$$\tau_{ext} = z_c \left(m\ddot{x} + B_d \left(\dot{x_0} - \dot{x} \right) + K_d \left(x_0 - x \right) \right) \tag{6.2}$$

where K_d and D_d are the desired stiffness and damping coefficients, respectively. The system oscillates around the equilibrium point x_0. By plugging (6.1) in (6.2), it is possible to obtain the desired CoM displacement as a function of the feedback torque τ_{fb}:

$$x_d(i) = \frac{\Delta t}{K_d \Delta t + B_d} A(i) + \frac{B_d}{K_d \Delta t + B_d} x_d(i-1) \tag{6.3}$$

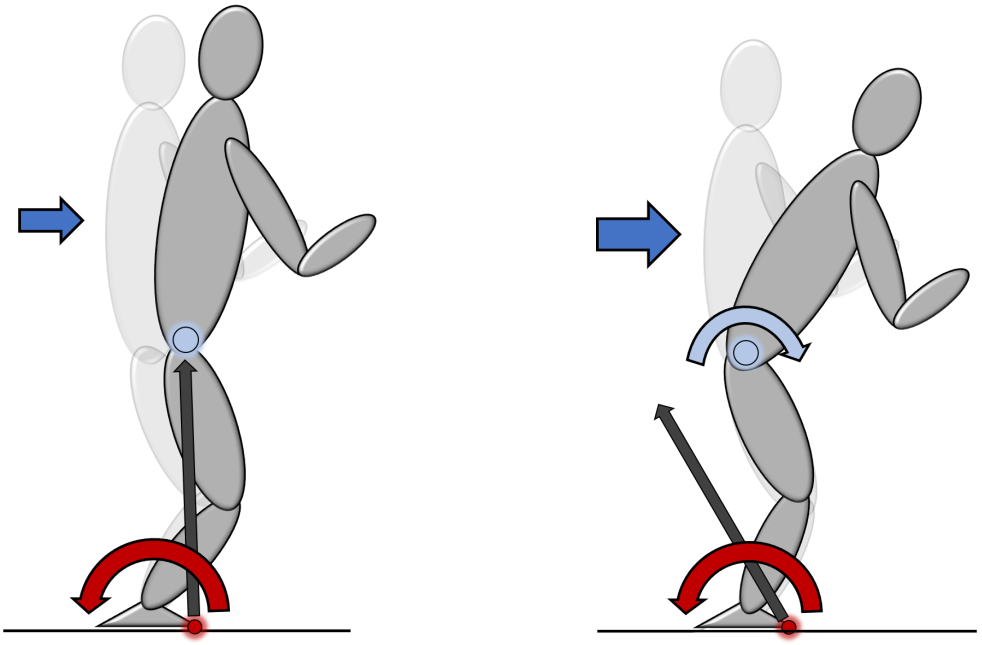
where:

$$\begin{aligned} A(i) =& K_d x_0(i) + B_d \dot{x_0}(i) + \frac{K_d - K_s}{z_c K_s} \tau(i) \\ &+ \frac{B_d}{z_c K_s} \dot{\tau}(i) - \frac{\tau_g(i)}{z_c} \end{aligned} \tag{6.4}$$

The active compliance is governed by the stiffness coefficient, while the damping dissipates the energy injected by disturbances. Indeed, these terms are variable, and they can be tuned depending on the parameters of the robot.

6.2.1 Integration

The contribution for this part was wrapping the controller in a plug-in compatible with the hybrid WPG architecture, and trying it both in simulation and on the real robot. As described in Section 5.7, the reference trajectories computed by the WPG are realized on the robot by the whole-body inverse kinematics solver CartesI/O (Laurenzi et al., 2019). It allows defining Cartesian Tasks and stack them in a SoT problem to set the desired priorities among its elements. For instance, one classical task is the control of the pose of a frame expressed w.r.t. another frame. In addition to "out of the box" tasks, CartesI/O allows the integration of custom tasks, that can be defined as plug-in and used in the SoT specification: implementing the compliant controller amounts to extend a CoM task to take into account the stabilizer reference, i.e. a desired CoM positional modification given the sensed reaction torque. By doing so, a CoM stabilizer is naturally integrated into the SoT, making it possible to set its priority w.r.t. to the other tasks, so that it does not interfere with higher priority tasks and it



(a) Ankle strategy for a biped, influences the linear momentum of the centroidal dynamics.

(b) Hip strategy for a biped, influences the angular momentum of the centroidal dynamics.

Figure 6.1 The ankle and hip strategies are bio-inspired motions that the robot uses to reject external pushes.

is guaranteed to run without any lower task interfering. On top of the possibility to control the position of the CoM, its reference is continuously adjusted by the compliant controller. Following its structure, the task exposes the stabilizer's relevant parameters, such as stiffness and damping, which corresponds to the level of compliance and passivity of the controller.

However, integrating the stabilizer into the control layer didn't meet the expected results, as the experiments on the real robot revealed how the robustness of the walking was not improved. This stabilizer only uses the CoP information to adjust the CoM, hence a more refined strategy for stabilization and disturbance rejection was attempted: the ankle and torso stabilizer.

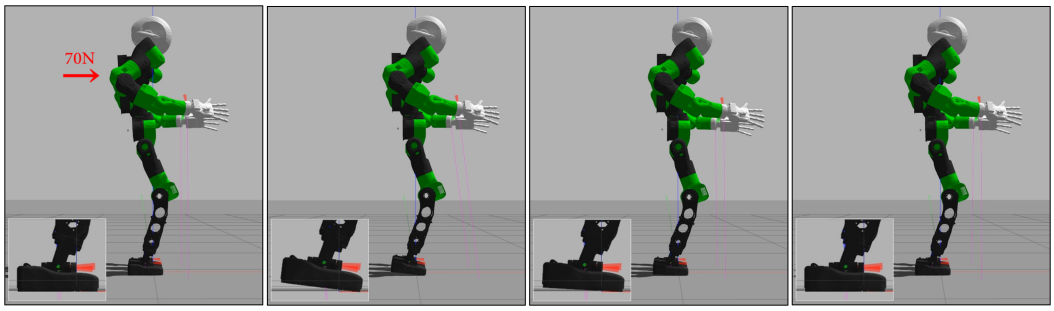
6.3 Ankle and Torso Stabilizer

In response to perturbation of the postural balance sensed by proprioception, the central nervous system triggers the activation of specific patterns in the muscular system. How the motor control is distributed along the muscular system depends on many external and internal factors, but generally distinct strategies circumscribed to discrete regions are preferred over

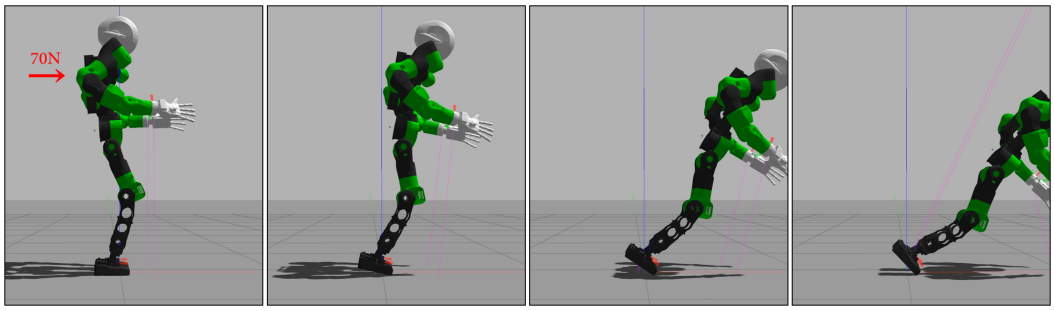
the continuous variation of muscular contractions. Among many reactions, involving joint rotations, limb motions or body adjustments, two regions of particular interest are mainly involved in postural equilibrium: the ankles and the hip. Intuitively, the ankle strategy consists in leveraging on the foot support to exert a torque about the ankle joint: this technique can be interpreted physically as a simple 1-DoF inverted pendulum actuated at the base joint, generating a torque limited by the lever arm of the supporting foot and the power of the tight and lower leg muscles. The hip strategy involves the rotation of the upper body, generating an angular moment along the sagittal or lateral axis passing through the hips, as studied in Runge et al. (1999). Given that the foot can only apply a pushing force through the sole in contact with the ground to obtain a torque about the ankle joint, due to the unilateral constraint, (see Chapter 2), the ankle strategy is used for recovery against slow and low amplitude disturbances. The hip strategy involves the whole upper body and the hips at the CoM level, so it can generate higher torques and consequently withstand higher and faster perturbations, as noticed in Nashner and McCollum (1985). Indeed, these recovery actions are usually employed in synergy, transitioning from one to the other depending on the amplitude of the disturbance.

Similar behaviours to the postural strategies observed in humans can be implemented on a humanoid robot. This can be achieved through different control schemes, but eventually these techniques boil down to direct manipulation of the system's momentum to compensate for external forces: in particular, the ankle strategy directly acts on the linear momentum of the centroidal dynamics, while the hip strategy modifies the angular momentum.

The chosen approach for the design of such controllers starts from the notion of the CP 2.5.5 and the approximation of the robot as a LIPM 2.5.3 to remove non-linear effects of the full dynamic model. The CP can be computed using (2.49), given an estimation of the CoM. This controller requires an observer to measure the real CoM pose and velocity. The observer is outside the scope of this thesis, and in the following presentation the sensed CP is considered as a given. Equation (2.51) and (2.50) decouple the CoM dynamics into one divergent and one stable component, respectively. The CP diverges from the CoP (or ZMP) while the CoM is drawn toward the CO. As already noticed in 2.5.5, stabilizing the CP is enough for postural balance recovery: therefore, by controlling the CoP in the support polygon, it is possible to regulate the CP which, in turns, naturally pulls the CoM. Indeed, the CP can be fully controlled only if it dwells inside the support polygon. If larger amplitude perturbations lead the CP outside the support, the ZMP travels to the edge of the region and the foot starts to rotate. In these cases, the robot can perform an action influencing the angular momentum or, if necessary, take a recovery step.



(a) The ankle controller is active. Responding to the external force, the ankles absorb the disturbance.



(b) The ankle controller is inactive. The external force pushes the robot out of balance.

Figure 6.2 Behaviour of COMAN+ when pushed by the same external force with and without the ankle controller.

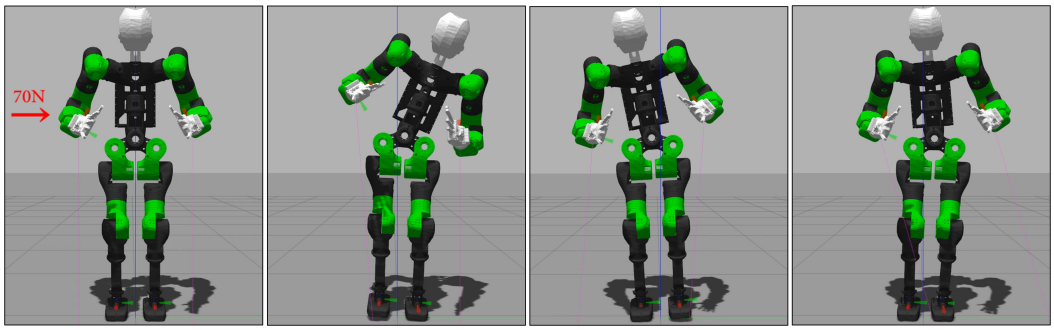
6.4 Contribution

A preliminary study is carried out to prove the effectiveness of the ankle-hip strategy. In particular, the simplest approach to regulate a variable given its reference is using a PD controller. The prototype of the ankle controller assumes a direct relationship between the CoP and the torque exerted by the ankle:

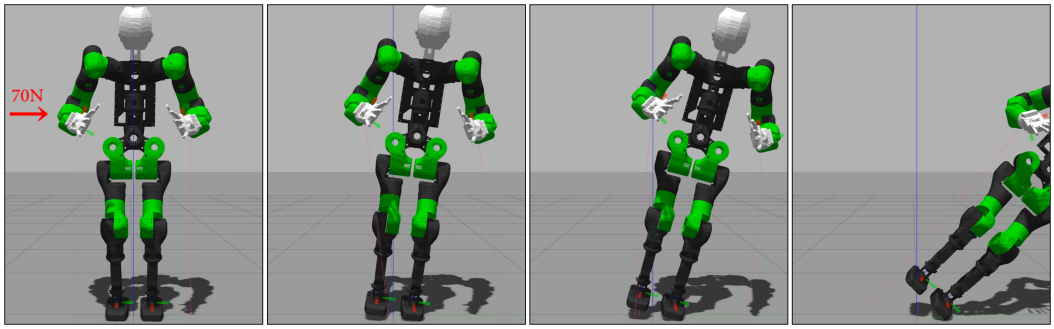
$$\boldsymbol{x}_{cop} = \frac{\boldsymbol{\tau}_{ankle}}{\boldsymbol{F}_{normal}} = \frac{\boldsymbol{\tau}_{ankle}}{m\boldsymbol{g}} \tag{6.5}$$

and the error on the CP is defined as $\tilde{\boldsymbol{x}}_{cp} = \boldsymbol{x}_{cp}^{ref} - \boldsymbol{x}_{cp}$, where $\boldsymbol{x}_{cp}^{ref}$ corresponds to the reference CP and $\boldsymbol{x}_{cp}$ is the CP computed using the sensed CoM. Consequently, the dynamics of the error on the CP can be written using (2.51) and plugging in (6.5):

$$\dot{\tilde{\boldsymbol{x}}}_{cp} = \omega \left(\tilde{\boldsymbol{x}}_{cp} - \frac{\boldsymbol{\tau}_{ankle}}{m\boldsymbol{g}} \right) \tag{6.6}$$



(a) The hip controller is active. Responding to the external force, the hip motion generate an angular momentum to counter the disturbance.



(b) The hip controller is inactive. The external force is high enough to disrupt postural balance.

Figure 6.3 Behaviour of COMAN+ when pushed by the same external force with and without the hip controller.

Taking as the control input the ankle torque, a simple PD controller can compensate the error of the CP, both in the lateral and the sagittal direction:

$$\begin{aligned} \tau_x &= k_p \tilde{\xi}_y + k_d \dot{\tilde{\xi}}_y \\ \tau_y &= k_p \tilde{\xi}_x + k_d \dot{\tilde{\xi}}_x \end{aligned} \tag{6.7}$$

Finally, the controller is implemented on the robot by commanding velocities on the robot instead of directly commanding the reference torques.

A similar approach leading to the same PD controller is used for the hip strategy: both strategies adjust the linear and angular momentum of the robot in response to changes in the CoP, resulting in disturbance rejection. The stabilizers are tested on the humanoid robot, comparing their effectiveness on COMAN+. Up to now, this combined strategy was only tried in simulation: while it showed extremely good results for rejecting external pushes when standing still, it is not robust when used for locomotion. Future works will involve running the controllers on the real robot to verify its efficacy. Furthermore, other strategies will be implemented to enhance the robustness of the walking, both around the WPG, to

stabilize an open-loop motion and embedded into the WPG, to change automatically step position to improve equilibrium.

6.5 Discussion

While the developed algorithms showed promising results, they didn't meet the expected results. A simple PD controller is enough for stabilizing the robot while standing still, but does not add robustness to locomotion. Future works will focus on designing a more refined control architecture for ankle-hip strategy, including a future reference as described in Aftab et al. (2012). Other methods will be tested, such as the chest posture controller, based on the work of Kajita et al. (2010), which computes angles for modification of the chest orientation.

Until now were discussed independent stabilizer added on top of the robot's control architecture. The other procedure will be targeted: embedding the stabilization layer into the WPG to modify relevant gait trajectories sent to the robot: changing step length and duration in response to external enhance the robustness of the gait, as described in Khadiv et al. (2017), or, following the scheme in Verrelst et al. (2006), cascading the PC in the hybrid WPG with another PC, forming a double preview controller that CoM corrections by feeding the previously computed ZMP trajectories in the same PC.

Chapter 7

Planning and Control Strategies for Physical Interaction Tasks

The two-fold problem of whole-body motion and environment interaction is tackled to achieve robust physical mobility and dexterity: an architecture for challenging physical interaction tasks is devised, addressing planning and control. Both levels are integrated into a framework for executing complex movements involving multi-contact scenarios. A particularly challenging high-level task is selected as a goal for COMAN+: the wall-plank, composed of unique movements that need to be planned and executed taking into account multiple contacts, balancing, and collisions.

To perform a general high-level task that comprises specific manoeuvres and multi-contact interaction with the environment, the pipeline for non-gaited multi-contact actions requires both a *planning* and a *control* layer. Both layers can select a suitable model of the humanoid robot according to the given task, choosing between two extremes: reduced models such as the Linear Inverted Pendulum (LIPM) on one side and whole-body descriptions on the other. This chapter presents a framework providing a full pipeline to execute a complex physical interaction behaviour of a humanoid bipedal robot, both from a theoretical and a practical standpoint. Building from a multi-contact control architecture that combines contact planning and reactive force distribution capabilities, the main contribution of this work consists in the integration of a sample-based motion planning layer conceived for transitioning movements where obstacle and self-collisions avoidance is involved. To plan these motions it is used a Rapidly Exploring Random Tree (RRT) projected on the contacts manifold and validated through the Centroidal Statics (CS) model, to ensure static balance on non-coplanar surfaces. Finally, the presented planning and control architecture is successfully validated on the humanoid robot COMAN+ performing a *wall-plank* task.

This chapter is based on the following article:
F. Ruscelli, M. P. Polverini, A. Laurenzi, E. M. Hoffman and N. G. Tsagarakis, "A Multi-Contact Motion Planning and Control Strategy for Physical Interaction Tasks Using a Humanoid Robot," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 25-29, 2020, Las Vegas, NV, USA (Virtual)

7.1 Introduction

As opposed to classical fixed-base robots, controlling free-floating platforms such as bipeds is considerably harder due to the inherent under-actuation that generates a strong coupling between physical interaction and body equilibrium. Humanoid research seeks to obtain dexterous and autonomous robots capable of mimicking the motion skills of a human being: in this respect, the two-fold problem of whole-body motion and environment interaction needs to be tackled to achieve robust physical mobility and interaction. A general high-level task is composed of unique movements that need to be planned and executed taking into account multiple contacts, balancing, and collisions. Hence, the building blocks of a controller for non-gaited multi-contact actions need to comprise a planning and a control layer, often organized in a hierarchical fashion (Kuindersma et al., 2016; Vaillant et al., 2016). Depending on the task, each layer adopts a suitable model of the humanoid robot such as the Centroidal Dynamics (CD) (Audren et al., 2014; Dai et al., 2014) or whole-body descriptions (Dalibard et al., 2013). The motion planning problem for humanoid robots is separated in *Cartesian-space planning*, which generates contact points and the relative interaction forces at each contact, and *Joint-space planning*, which connects the sequence of planned stances with a continuous motion without violating a set of constraints. Two approaches to the planning problem can be found in literature: sample-based and gradient-based. The first uses discrete search methods such as A* or RRT (Escande et al., 2013; Hauser et al., 2008; Nishi and Sugihara, 2014). While being a powerful tool, raising the complexity of the model greatly impacts the computation time for query a solution, and a guiding heuristic to span the search space may not be trivial to be determined, since the majority of the randomly sampled configurations do not satisfy all the constraints. The second relies on continuous optimization to find feasible solutions. It avoids unnatural motions and, if fed with the full dynamic model, produces refined behaviours (Mordatch et al., 2012), but its computational complexity makes it unpractical for many real-world scenarios. The problem can be also formulated by using models such as the CD, trading dynamic consistency for efficiency. To compute the contacts and the interaction forces, a gradient-based method was exploited, using the

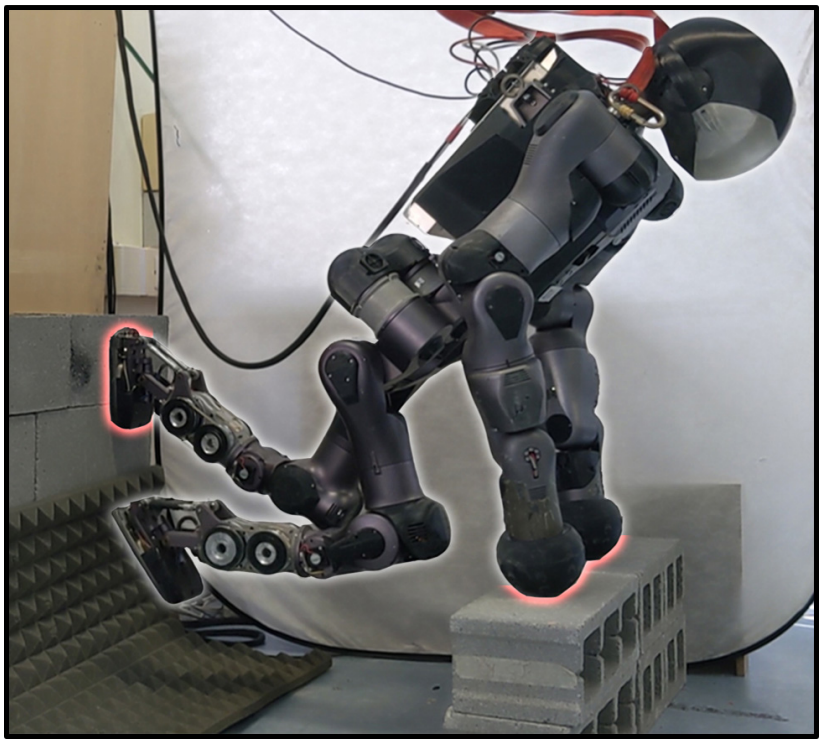


Figure 7.1 COMAN+ reaching the *wall plank* pose. The weight of the robot is distributed among the hands and the left foot pushing against the wall.

robot CD under quasi-static assumptions, valid for any task which involves quasi-static poses. During the motion towards the planned contact, preventing joint limits, (self-)collisions and singular configurations while ensuring kinematic reachability and stability condition is critical, especially for a humanoid robot moving in a confined space. A RRT algorithm was employed, using the full kinematic model of the robot to query a suitable joint-space in the heavily constrained configuration space.

Motion control is responsible for the closed-loop tracking of the trajectories and the interaction forces planned. In executing the planned trajectories, this layer provides compliance and robustness, critical requirements for the balancing of the robot. Two categories of balancing controllers exist: the first directly relies on full-body inverse dynamics (Posa et al., 2016; Saab et al., 2013; Tassa et al., 2012) to solve the optimal force distribution problem. The second approximates the model used in the previous methods: it is based on a two-stage approach which involves either a pre-optimization of the contact forces (Henze et al., 2016; Lee and Goswami, 2010; Werner et al., 2016) or a post-optimization (Laurenzi et al., 2018) before mapping them to joint torques. The *Interaction controller* is based on the last approach.

The contribution of this work lies in the development of a system for planning and control of multi-contact interaction tasks for hyper-redundant floating-base robots and its implementation on the humanoid COMAN+ (see Figure 7.1). In particular, the work in Parigi Polverini et al. (2020) is extended by providing:

- a sample-based planner to generate a sequence of feasible robot configurations, in terms of (self-)collisions and balance, from a multi-contact stance to another;
- an optimization-based interpolation of the planned sequence to obtain a time-optimal joint trajectory satisfying velocity and acceleration constraints.

Each layer is integrated into a framework that allows executing complex motion actions while guaranteeing quasi-static equilibrium. Despite great progression in simulation, these complex physical interaction tasks are still difficult to replicate in the real world: this work takes a step towards bridging the gap between simulation and real-hardware achievements by successfully validating the proposed control architecture on the robot COMAN+. It was chosen a challenging action for the average human being that consists in putting both hands on a support while rising both feet on a wall and subsequently balancing by distributing its weight on the four limbs keeping the whole body suspended from the ground, from here on referred as the *wall plank*. The designated motion highlights the critical role that contact interactions play in any motion that goes beyond nominal walking. Furthermore, it is highly non-linear and it is performed in a confined space, making it particularly challenging for the geometry of the robot, which does not have the full flexibility and mobility of a human being.

7.2 Mathematical background

7.2.1 Floating-base dynamic model

Humanoid robots fall in the category of hyper-redundant under-actuated robots. The vector of generalized variables of the n-link robot is composed of an un-actuated and an actuated part, as described by equation 2.10, where $\boldsymbol{q}_{\text{base}} \in \boldsymbol{R}^6$, which is the configuration of the (un-actuated) virtual chain describing the pose of the robot w.r.t. to an inertial world frame, while $\boldsymbol{q}_{\text{joint}} \in \boldsymbol{R}^n$ represents the coordinates of the n actuated joints. As introduced in Chapter 2, the model describing the robot dynamics can be written in the following form:

$$\begin{bmatrix} \boldsymbol{B}_u(\boldsymbol{q}) \\ \boldsymbol{B}_a(\boldsymbol{q}) \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{h}_u(\boldsymbol{q},\dot{\boldsymbol{q}}) \\ \boldsymbol{h}_a(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{6\times n} \\ \boldsymbol{I}_{n\times n} \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \boldsymbol{J}_{\boldsymbol{C},u}^{\top} \\ \boldsymbol{J}_{\boldsymbol{C},a}^{\top} \end{bmatrix} \boldsymbol{F}_{\boldsymbol{C}} \tag{7.1}$$

which entails a selection matrix that differentiate actuation torques $\boldsymbol{\tau}$ from the full set of torques of the model. The unilateral contact forces $\boldsymbol{F_C} \in \mathbb{R}^k$ are mapped in the joint-space by the Jacobian of all the links in contact with the environment $\boldsymbol{J}_\mathrm{C} \in \mathbb{R}^{k\times(n+6)}$ In the case of full planar contact $k = 6$, while for a point contact $k = 3$.

7.2.2 Reduced model

Full dynamic models grant kinematic and dynamic consistency at the expenses of a higher computational complexity. Reduced methods, on the other hand, rely on a lighter description: the most commonly used are the CD model, presented in 2.4.3 or the CS, a simplified version which discards acceleration and velocity. The CD model simplifies the full model by projecting the first six rows of (7.1) on the robot CoM:

$$\boldsymbol{A}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{A}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = m\boldsymbol{g} + \boldsymbol{G}_\mathrm{CD}\boldsymbol{F_C} \tag{7.2}$$

where $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{6\times(n+6)}$ is the *centroidal momentum matrix* (CMM). Terms $\boldsymbol{g} \in \mathbb{R}^3$ and $m \in \mathbb{R}$ are the vector of the gravity acceleration and the mass of the robot, respectively. Finally, $\boldsymbol{G}_\mathrm{CD} \in \mathbb{R}^{6\times k}$ is the the centroidal dynamics grasp matrix:

$$\boldsymbol{G}_\mathrm{CD} = \begin{bmatrix} \boldsymbol{I}_3 & \cdots & \boldsymbol{I}_3 \\ \boldsymbol{S}(\boldsymbol{r}_\mathrm{CoM} - \boldsymbol{r}_{\mathrm{C},1}) & \cdots & \boldsymbol{S}(\boldsymbol{r}_\mathrm{CoM} - \boldsymbol{r}_{\mathrm{C},n_\mathrm{C}}) \end{bmatrix} \tag{7.3}$$

where n_C is the number of contacts and $\boldsymbol{r}_{\mathrm{C,i}} \in \mathbb{R}^3$ is the i-th contact position, while $\boldsymbol{S}$ is a skew-symmetric matrix. For any application in which only static equilibrium is required, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ can be discarded for a further simplification. Under quasi-static conditions, (7.4) yields the formulation of the CS:

$$m\boldsymbol{g} + \boldsymbol{G}_\mathrm{CD}\boldsymbol{F_C} = \boldsymbol{0} \tag{7.4}$$

This reduced description assumes fixed contact placements with associated friction models to obtain each interaction force required to compensate for gravity and achieve static balancing.

7.2.3 Coulomb friction cones

Deriving contact-stability conditions for the robot requires to model the interaction between the robot and the environment. The normal and tangential component of the interaction forces are defined as in (2.46). The same equation holds for any contact point C_i with the

associated normal to the surface $\boldsymbol{n}_{\boldsymbol{C},i} \in \boldsymbol{R}^3$:

$$\begin{aligned} \boldsymbol{F}^n_{\boldsymbol{C},i} &= \left(\boldsymbol{F}_{\boldsymbol{C},i} \cdot \boldsymbol{n}_{\boldsymbol{C},i}\right) \boldsymbol{n}_{\boldsymbol{C},i} \\ \boldsymbol{F}^\top_{\boldsymbol{C},i} &= \boldsymbol{F}_{\boldsymbol{C},i} - \left(\boldsymbol{F}_{\boldsymbol{C},i} \cdot \boldsymbol{n}_{\boldsymbol{C},i}\right) \boldsymbol{n}_{\boldsymbol{C},i} \end{aligned} \tag{7.5}$$

As presented in 2.5.4, a contact is guaranteed as long as $\boldsymbol{F}_{\boldsymbol{C},i}$ lies inside the friction cone directed by $n_{\boldsymbol{C},i}$:

$$\mathcal{F}\left(\boldsymbol{F}_{\boldsymbol{C},i}, \boldsymbol{n}_{\boldsymbol{C},i}, \mu_i\right) := \begin{cases} \boldsymbol{F}_{\boldsymbol{C},i} \cdot \boldsymbol{n}_{\boldsymbol{C},i} > F_{\text{thr}} \\ \left\| \boldsymbol{F}^t_{\boldsymbol{C},i} \right\|_2 \leq \mu_i \left(\boldsymbol{F}_{\boldsymbol{C},i} \cdot \boldsymbol{n}_{\boldsymbol{C},i}\right) \end{cases} \tag{7.6}$$

where μ_i is the static friction coefficient at the contact point C_i. The Euclidean norm $\| \cdot \|_2$ simplifies the model, by assuming a circular friction cone, i.e. the isotropy of friction.

7.3 Method overview

The proposed strategy extends the work in Parigi Polverini et al. (2020): it is divided into a motion planning layer, which accounts for contact position, interaction forces, joint-space planning and a motion control layer that realizes the planned trajectory while granting continuous multi-contact equilibrium of the robot. Its hierarchical layered architecture is illustrated in Figure 7.3:

- *Cartesian-space planner*: prescribes reference values for contact force distribution $\boldsymbol{F}_{\boldsymbol{C},i}$ to achieve static equilibrium. It also outputs the position of each contact $\boldsymbol{r}_{\boldsymbol{C},i}$ and CoM $\boldsymbol{r}_{CoM}$, given the desired values. The solution is found via a non-linear optimization which reasons about the CS of the robot to reduce the problem complexity.

- *Joint-space planner*: generates a suitable joint trajectory $\overline{\boldsymbol{q}}(t)$ to reach the desired contact without violating balancing constraints and avoiding kinematic singularities as well as collisions with the environment and the robot itself.

The planned references are tracked by the motion control layer: planned trajectories can be either directly realized in joint-space or provided by a Cartesian controller based on Hierarchical Inverse Kinematics (HIK) when the joint-space planner is not required. The *Interaction controller* continuously distributes the planned contact force on each contact end-effector, generating feed-forward torque references $\boldsymbol{\tau}_{\text{ff}}$ for the joint-level control loop. Both feet and hands are required to reach for a surface and establish a robust contact: in

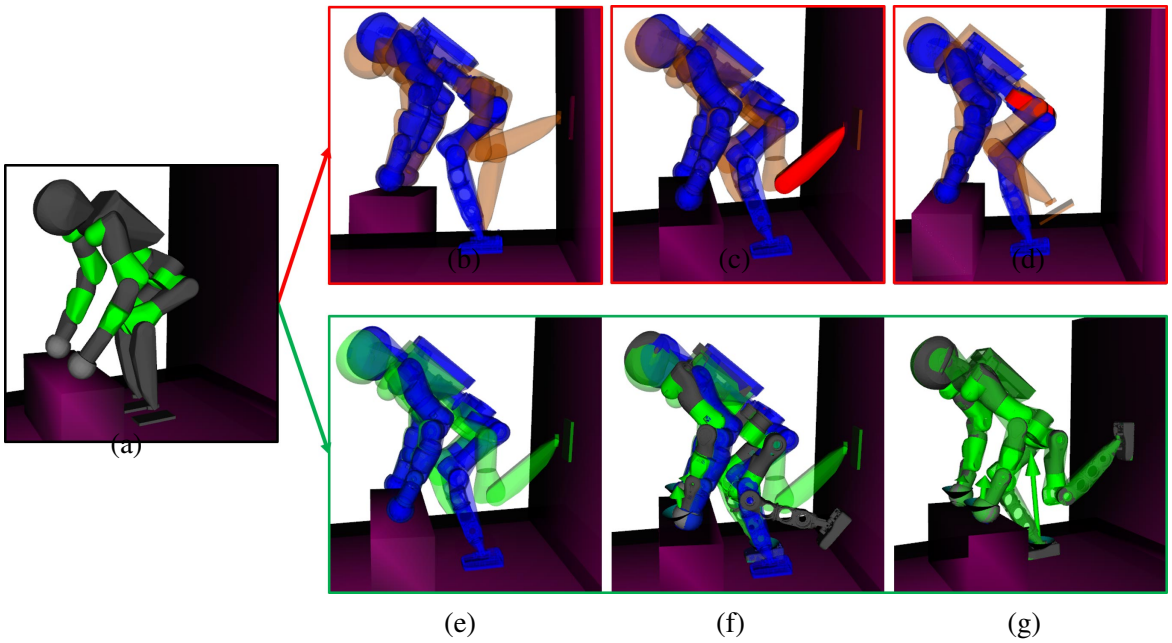


Figure 7.2 Simulations of COMAN+ positioning a foot on the wall.

doing so, a collision detection algorithm senses if the controlled end-effector is in contact with the environment according to a prescribed threshold.

7.4 Motion planning

This section presents the strategy for planning environment interactions: a gradient-based method efficiently computes the Cartesian position of the contacts and the forces required at each supporting link for static equilibrium. To reach the desired contacts a set of feasible configurations of the robot is found by a sample-based algorithm, particularly suited when obstacle and self-collision avoidance are involved. Finally, an optimization problem interpolates the planned sequence of configurations ensuring a joint trajectory with bounded accelerations.

7.4.1 Cartesian-space planner

To grant quasi-static contact stability for a given robot configuration a critical role is played by the interaction forces. The first layer relies on non-linear programming (NLP) to compute force distribution among the subset of end-effector in contact with the environment. Additionally, given a CoM position reference, it returns a feasible CoM position and the active

contact positions that satisfy the balancing conditions. At each stance the algorithm is fed with:

- *active contacts* $\boldsymbol{C}_a$, specifying each contact position $\boldsymbol{r}_{\boldsymbol{C},a}$ and its associated normal $\boldsymbol{n}_{\boldsymbol{C},a}$;
- *lifted contacts*, $\boldsymbol{C}_l$, enforcing the constraint $\boldsymbol{F}_{\boldsymbol{C},l} = 0$, to redistribute the robot weight on the remaining contacts. Note that, if no lifted contact are required, constraint (7.7c) not necessary;
- optionally, a desired CoM position $\boldsymbol{r}_{\text{CoM}}^{\text{des}}$.

The resulting NLP in the variables $\boldsymbol{x} = \left[\boldsymbol{r}_{\text{CoM}}, \boldsymbol{r}_{\boldsymbol{C},a}, \boldsymbol{F}_{\boldsymbol{C}}\right]$ is formulated as:

$$\min_{\boldsymbol{x}} \|\boldsymbol{F}_{\boldsymbol{C}}\|^2_{2,W_1} + \|\boldsymbol{r}_{\text{CoM}} - \boldsymbol{r}_{\text{CoM}}^{\text{des}}\|^2_{2,W_2} + \|\boldsymbol{r}_{\boldsymbol{C},a} - \boldsymbol{r}_{\boldsymbol{C},a}^{des}\|^2_{2,W_3}$$

$$\begin{aligned}
\text{s.t.:} \quad & m\boldsymbol{g} + \boldsymbol{G}_{\text{CD}}\boldsymbol{F}_{\boldsymbol{C}} = \boldsymbol{0} && \text{(7.7a)}\\
& \left\{\boldsymbol{F}_{\boldsymbol{C},a}, \boldsymbol{n}_{\boldsymbol{C},a}\right\} \in \mathcal{F}\left(\boldsymbol{F}_{C,a}, \boldsymbol{n}_{\boldsymbol{C},a}, \mu_a\right) && \text{(7.7b)}\\
& \boldsymbol{F}_{C,l} = 0 && \text{(7.7c)}\\
& \boldsymbol{r}_{\text{C},i} \in S_{\text{C}}(\boldsymbol{r}_{\text{C},i}) && \text{(7.7d)}
\end{aligned}$$

Constraint (7.7a) ensures quasi-static stability based on the CS of the robot, while (7.7b) guarantees contact-stable stances given desired contact position computing the friction cone $\mathcal{F}$ as a function of the contact positions $\boldsymbol{r}_{\boldsymbol{C},a}^{\text{des}}$, the corresponding normals $\boldsymbol{n}_{\boldsymbol{C},a}^{\text{des}}$ and the given friction coefficient μ_a. Constraints (7.7c) set to zero the required force at the lifted contact l. Finally, constraint (7.7d) requires the contact to lay on the environment. To get the benefit of continuous optimization, a significantly faster method compared to Mixed Integer Programming where computational complexity rapidly increases as the number of decision variables grows, a continuous description of the environment is formulated through superquadric functions $S_{\boldsymbol{C}}(\boldsymbol{r}_{\boldsymbol{C},i})$, as proposed in Parigi Polverini et al. (2020).

7.4.2 Joint-space planner

This layer is in charge of generating feasible transition motions between the static stances. However, these transitions are highly non-linear and constrained, as highlighted by the experiments in simulation shown in Figure 7.2: the robot incurs in unfeasibility (in red) such as kinematic singularities, e.g. full extension of the knee joint of the supporting leg

(b), collisions with the environment, e.g. foot impacting the wall (c) or self-collision, e.g. hip joints colliding (d). As a matter of fact, a standard HIK incurs in local minima and should be augmented with obstacle, self-collision and singularities avoidance. To prevent these issues, a naive approach is tentatively selected, to validate whether two stances can be connected: a sequence of Cartesian end-effector sub-poses are hand-picked to achieve the final configuration. Although viable, this approach is burdensome and highly specific, since it requires fine-tuning for each different robot action. For these reasons, the choice fell on a more general approach that accounts for environmental and balancing constraints as well as self-collision and kinematic singularities. Sample-based planner such as RRT are particularly suited for this task:

- self-collision are rapidly checked and the relative joint configurations discarded;
- contacts can be guaranteed by projecting the robot configuration on the desired manifold;
- validity checks can be applied to impose stability constraints and (self-)collision free paths.

The proposed planner uses a probabilistically complete sample-based algorithm to find a feasible solution. Given the starting configuration $\boldsymbol{q}_{start}$ and final configuration $\boldsymbol{q}_{goal}$, the planner finds a collision-free joint trajectory: since the motion is realized in joint-space, kinematic singularities do not arise. Contact consistency, due to the floating-base nature of the humanoid robot, is injected in the problem using manifold projection: robot configurations $\boldsymbol{q}_i$ are mapped on an implicit manifold (Şucan et al., 2012) defined by the constraint function $F(\boldsymbol{q}) = 0$ that describes a set of end-effectors in contact with the the environment. Physical limitations, such as balancing, are subsequently considered by validating the projected configuration. In the general case of multiple non-coplanar contacts, where the support polygon does not suffice to grant static equilibrium, the following optimization problem in the contact force variables is formulated:

$$\begin{aligned} \min_{\boldsymbol{F_C}} \ & \left\| \boldsymbol{J}_{\boldsymbol{C},u}^{\top} \boldsymbol{F_C} - \boldsymbol{g}_u(\boldsymbol{q}) \right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{b}_l \leq \boldsymbol{D}\boldsymbol{R}_{\boldsymbol{C}} \boldsymbol{F_C} \leq \boldsymbol{b}_{\text{u}} \end{aligned} \tag{7.8}$$

which corresponds to enforcing the CS (7.4) at the floating base, under linearized friction cones constraints. Notice that the problem as formulated in (7.8) has always a solution inside the friction cones. The projected pose is considered statically balanced if the cost function in (7.8) does not exceed a certain threshold ε, and, if collision-free, the pose is added to the

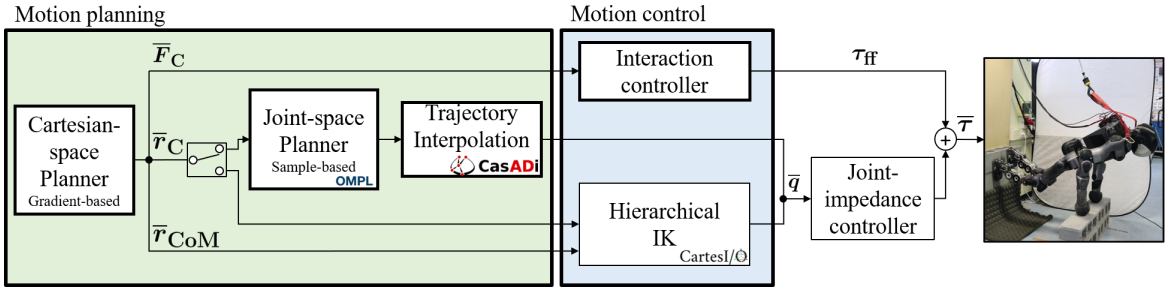


Figure 7.3 Scheme of the proposed multi-contact motion planning and control architecture.

search tree. Any configuration that violates the CS is discarded. The procedure exploits the OMPL library (Şucan et al., 2012) to set up the randomized planning problem and spans the constrained configuration space of the humanoid until it reaches a feasible goal (see Algorithm 1). The planning scene consists of all the obstacles involved, to avoid any stance in collision with the environment. The robot configuration is projected on the manifold using an atlas-based methodology (Şucan et al., 2012). Self-collision are detected using the FCL package (Pan et al., 2012). To speed up the algorithm, all possibly colliding pairs of links are identified to discard unnecessary checks for self-colliding configurations. In Figure 7.2, the initial quadrupedal pose is shown in (a), fed to the Joint-space planner as the starting configuration $\boldsymbol{q}_{start}$ (in blue). A feasible configuration is green. The Joint-space planner checks if the goal configuration $\boldsymbol{q}_{end}$ is feasible, as shown in (e) and produces a joint trajectory from $\boldsymbol{q}_{start}$ to $\boldsymbol{q}_{goal}$ without violating the CS constraints. Note in (f) and (g) the friction cones and the relative contact forces satisfying the CS. Future works will allow specifying solely the contacts to automatically get a feasible goal configuration instead of providing a pre-planned one. The final set of stances found by the planner is then interpolated to get a joint trajectory.

7.4.3 Trajectory interpolation

Because the selected RRT-based planner is a *geometric* planner, its output is a simple sequence of points $\{\boldsymbol{q}_k\}_{k=0}^{N}$ in the robot configuration space, that does not carry any information about its time parametrization. Consequently, there is no guarantee whatsoever that output points will accumulate in correspondence to high-acceleration trajectory segments. The direct interpolation of the output sequence assuming an evenly-spaced time grid will therefore lead to high-acceleration, jerky motions, especially when changes of directions occur to any of the robot degrees of freedom. The chosen solution to address this issue was performing an

optimization-based interpolation relying on polynomial splines of the following form:

$$\boldsymbol{q}(t) = \boldsymbol{s}_k(t), \quad t \in [t_k, t_{k+1}], \; k = 0, \ldots, N-1 \tag{7.9}$$

$$\boldsymbol{s}_k(t) = \sum_{i=0}^{d} \boldsymbol{a}_{k,i}(t - t_k)^i, \tag{7.10}$$

where (i) $\boldsymbol{a}_{k,i} \in \mathbb{R}^n$ are the spline coefficients, (ii) $d \in \mathbb{N}^+$ is the spline degree, and (iii) $t_k \in \mathbb{R}$ is the k-th time knot. The spline degree is fixed to $d = 3$ in this work, which yields a continuous velocity profile with bounded acceleration.

Algorithm 1: Multi-Contact Motion Planning and Control

Data: $\boldsymbol{q}_{goal}, \boldsymbol{C}_a, \boldsymbol{C}_l, \boldsymbol{r}^{des}_{\boldsymbol{C},a}$, obstacles
Result: collision-free, statically balanced motion
for *multi-contact action in task* **do**
1 set $\boldsymbol{C}_l$ to Cartesian-space Planner
2 run Cartesian-space Planner
→ get $\{\overline{\boldsymbol{r}}_{\boldsymbol{C},a}, \overline{\boldsymbol{r}}_{CoM}, \overline{\boldsymbol{F}}_{\boldsymbol{C},a}\}$
3 send $\overline{\boldsymbol{r}}_{CoM}$ to HIK
4 set up *Planning Scene*
5 set $\boldsymbol{q}_{start}$ to Joint-space Planner
6 generate *contact manifold* from $\{\boldsymbol{q}_{start}, \boldsymbol{C}_a\}$
7 set $\boldsymbol{q}_{goal}$, project it on the *contact manifold*
while *(exact solution found)* **do**
8 run RRT constrained with CS → get $\boldsymbol{q}_k$
end
9 optimal trajectory interpolation: $\boldsymbol{q}_k \rightarrow \boldsymbol{q}(t)$
10 lift contact $\boldsymbol{C}_l$ with HIK
11 send $\boldsymbol{q}(t)$ to robot
12 run HIK for surface-reaching
13 add new contact $\boldsymbol{C}_a$ to Cartesian-space Planner
14 send $\boldsymbol{F}_{\boldsymbol{C},a}$ to Interaction Controller
end

With the aim to compute a *time-optimal* trajectory satisfying any given joint velocity and acceleration bounds, the following optimization problem is formulated:

$$\min_{\boldsymbol{a},\Delta\boldsymbol{t}} \quad \mathbf{1}^\top \Delta\boldsymbol{t} + \lambda \|\boldsymbol{a}\|^2 \tag{7.11}$$

$$\text{s.t.:} \quad -\dot{\boldsymbol{q}}_{\max} \leq \dot{\boldsymbol{s}}_k(t_k) \leq \dot{\boldsymbol{q}}_{\max} \quad \forall k = 0,\ldots,N \tag{7.12}$$

$$-\ddot{\boldsymbol{q}}_{\max} \leq \ddot{\boldsymbol{s}}_k(t_k) \leq \ddot{\boldsymbol{q}}_{\max} \quad \forall k = 0,\ldots,N \tag{7.13}$$

$$\boldsymbol{s}_{k-1}(t_k) = \boldsymbol{s}_k(t_k) = \boldsymbol{q}_k \quad \forall k = 1,\ldots,N \tag{7.14}$$

$$\dot{\boldsymbol{s}}_{k-1}(t_k) = \dot{\boldsymbol{s}}_k(t_k) \quad \forall k = 1,\ldots,N \tag{7.15}$$

$$\Delta\boldsymbol{t} \geq \mathbf{0} \tag{7.16}$$

where the minimization of (7.11) is carried out over the vector of all spline coefficients $\boldsymbol{a} \in \mathbb{R}^{n \cdot d \cdot N}$ and the vector of time interval *durations* $\Delta\boldsymbol{t} \in \mathbb{R}^N$, whereas the constraints (7.12)-(7.16) enforce velocity and acceleration limits, plus the interpolation and continuity conditions, and positive interval durations. In order to regularize the solution, the spline coefficients norm is penalized via a small scalar $\lambda > 0$. The resulting non-linear problem is high-dimensional but highly sparse as well, and it can be solved in a fraction of the time required by the sample-based planner.

7.5 Interaction control

This controller is based on (Laurenzi et al., 2018), which post-optimizes the contact forces to account for the under-actuated component of the system. As depicted in Figure 7.3, it consists in feeding the inner joint impedance control loop:

$$\boldsymbol{\tau} = \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q}) + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_{\text{ff}} \tag{7.17}$$

the feed-forward torque:

$$\boldsymbol{\tau}_{\text{ff}} = -\boldsymbol{J}_{\boldsymbol{C},\text{a}}^\top \boldsymbol{F}_{\boldsymbol{C}} \tag{7.18}$$

The reference contact force is obtained solving an optimization problem to track the planned contact force $\overline{\boldsymbol{F}}_{\boldsymbol{C}}$ under two conditions for balancing:

$$\begin{gathered} \min_{\boldsymbol{F}_{\boldsymbol{C}}} \left\| \boldsymbol{F}_{\boldsymbol{C}} - \overline{\boldsymbol{F}}_{\boldsymbol{C}} \right\|_2^2 \\ \text{s.t.} \\ \boldsymbol{J}_{\boldsymbol{C},u}^\top \boldsymbol{F}_{\boldsymbol{C}} = \boldsymbol{g}_u(\boldsymbol{q}) \\ \boldsymbol{b}_l \leq \boldsymbol{D}\boldsymbol{R}_{\boldsymbol{C}} \boldsymbol{F}_{\boldsymbol{C}} \leq \boldsymbol{b}_{\text{u}} \end{gathered} \tag{7.19}$$

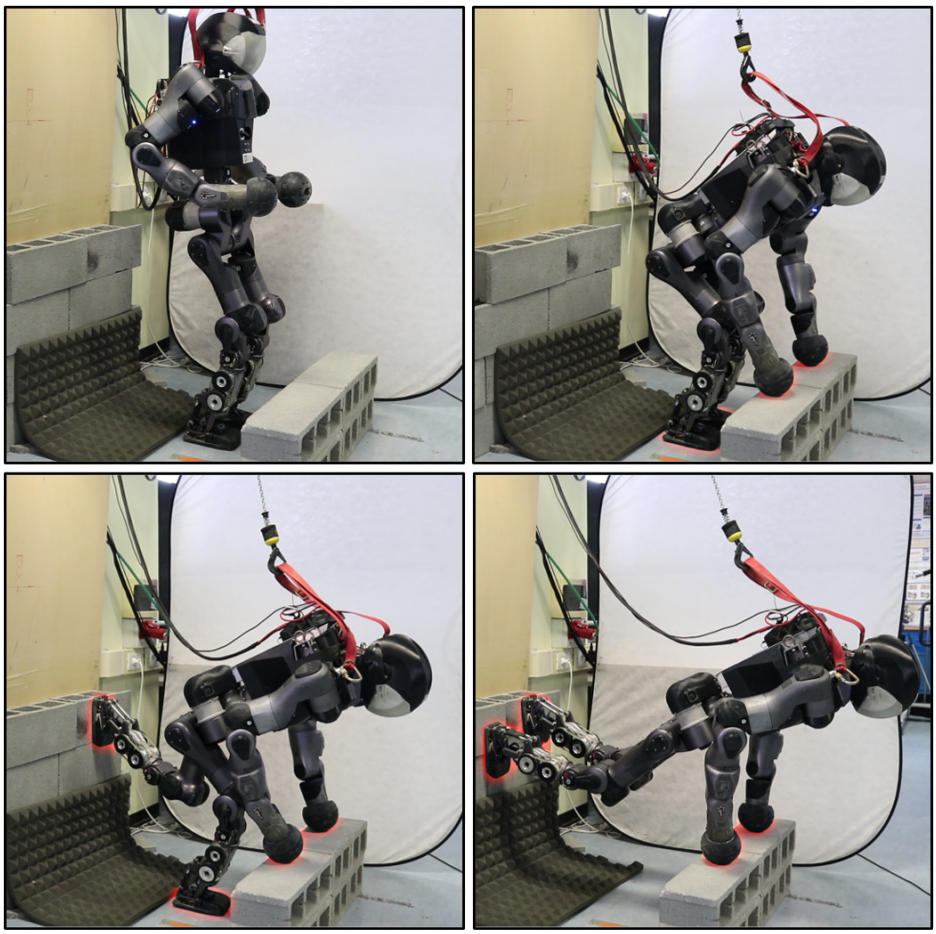


Figure 7.4 Sequence of frames of COMAN+ performing the *wall plank*.

The first equality constraint enforces centroidal statics balancing (7.4). The second constraint bounds the contact forces to satisfy the friction cones contact condition. In particular, the matrix $\boldsymbol{R_C}$ accounts for the contact orientation: given the normal to the surface in the contact point $\boldsymbol{C}$, a rotation matrix $\boldsymbol{R_C}$ can be found such that ${}^{\boldsymbol{C}}\boldsymbol{F_C} = \boldsymbol{R_C}\boldsymbol{F_C}$, where ${}^{\boldsymbol{C}}\boldsymbol{F_C} \in \mathbb{R}^3$. The matrix $\boldsymbol{D}$ is related to the static friction coefficient:

$$\begin{aligned} {}^{\boldsymbol{C}}\boldsymbol{F}_{\boldsymbol{C}}^{z} &\geq 0, \\ \left| {}^{\boldsymbol{C}}\boldsymbol{F}_{\boldsymbol{C}}^{x,y} \right| &\leq \tilde{\mu}\, {}^{\boldsymbol{C}}\boldsymbol{F}_{\boldsymbol{C}}^{z} \end{aligned} \tag{7.20}$$

where $\tilde{\mu} = \frac{\sqrt{2}}{2}\mu$ is an inner approximation for the isotropic Coulomb friction cone in (7.6). To recover the contact forces the Jacobian $\boldsymbol{J}_{\boldsymbol{C},u}^{\top}$ and the gravitational term $\boldsymbol{g}_u(\boldsymbol{q})$ are computed in closed-loop using the robot model and the IMU measurements.

7.6 Experimental validation

In order to show the capabilities of the proposed architecture, an uncommon and challenging task for a bipedal humanoid was conceived, the *wall plank*: the task consists in reaching a pose with two hands close to the ground level and two feet on an adjacent wall in the minimum number of actions, starting from a standing position. Algorithm 1 describes a general procedure that is exploited to execute this particular physical interaction task.

7.6.1 Wall plank

The robot crouches to reach a quadrupedal stance with each hand resting on a pile of two bricks (at 30 cm from the ground, on a surface of 50x20 cm). The distance between the middle of the robot soles and the wall is approximately 65 cm, as shown in Figure 7.1. In the initial configuration the robot is standing on its feet, while in the goal position the body is held slanted to the ground, supported by the hands and pushing on the wall with its feet. The robot weighs 70 kg and is 1.70 m tall. The overall motion is composed of strongly non-linear manoeuvres due to the robot kinematics, the environment geometry and the inherently cramped poses the robot needs to adopt. Hence a strong emphasis is put on generating feasible trajectories that satisfy joint limits and avoid self-collisions and singularities. In particular, to reach the goal position the robot goes through a discrete sequence of multi-contact stances:

- *Standing stance*: initial configuration, the robot is standing on its feet;
- *Crouched stance*: bent, with feet on the ground and hands on two piled bricks;
- *Single leg stance*: one leg is lifted on the wall;
- *Double leg stance*: goal configuration, both feet are on the wall while the hands are on the bricks.

The robot performing the above-mentioned task is COMAN+, described in Chapter 3. In particular, each joint is equipped with a torque sensor, enabling the implementation of torque and impedance control, while standard position control is also available. A force/torque sensor is positioned on each foot. The soles are covered with rubber, while the anthropomorphic hands are replaced with rubber-coated spherical end-effectors. Their extremities are sensor-less, but the resulting configuration is particularly suited for a task in which ruggedness and high friction are essential.

7.6.2 Method specifics

Trajectory generation for the transition to a quadrupedal stance relies on a whole-body hierarchical inverse kinematics framework named CartesI/O (Laurenzi et al., 2019) to define multiple Cartesian tasks organized in a SoT fashion. For simple trajectories, this local method is preferred, since it is efficient and deterministic. However, the last two poses entail a highly non-linear motion in a narrow space, limited by environmental obstacles and kinematic constraints (i.e. the last 2 DoF of the robot are situated in the torso, allowing rotation only around the longitudinal and sagittal axis): the HIK, being a local method, runs up against ill-behaviours such as singularities and self-collisions, detrimental in these delicate phases. The integration of joint-space and Cartesian-space control enhances the framework adaptability: straightforward behaviours such as CoM shifting and surface-reaching are managed by CartesI/O while complex motions are autonomously planned and sent to the robot as joint-space trajectories. The proposed framework is organized as follows:

- the *Cartesian-space planner* is based on IFOPT, an Eigen-based C++ interface to the NLP solver IPOPT (Wächter and Biegler, 2006). At each stage, it produces the contact forces and the end-effector (contacts) positions to achieve static equilibrium, given the desired CoM and end-effector reference positions and the associated bounds.
- the *Joint-space planner* runs the RRT-Connect algorithm based on the OMPL library (Şucan et al., 2012) to find a suitable set of robot configuration for transitioning from one stance to another without violating the multiple constraints. The interpolation between the resulting configurations is formulated as an optimal control problem using CasADi (Andersson et al., 2019) which generates a NLP solved with IPOPT.
- the *Interaction controller* is responsible for the force distribution problem, running in a ROS node at 100 Hz. The forces at each supporting link are computed using the closed-loop joint coordinates and the IMU readings from the robot.

CartesI/O runs on separate node at 100 Hz, and a specific SoT was tailored for this work:

$$\begin{pmatrix} \left({}^{\mathrm{W}}\mathcal{T}_{\mathrm{LFoot}} + {}^{\mathrm{W}}\mathcal{T}_{\mathrm{RFoot}} + {}^{\mathrm{W}}\mathcal{T}^{RPY}_{\mathrm{Torso}}\right) / \\ {}^{\mathrm{Waist}}\mathcal{T}_{\mathrm{LArm}} + {}^{\mathrm{Waist}}\mathcal{T}_{\mathrm{RArm}} / \\ {}^{\mathrm{W}}\mathcal{T}^{XY}_{\mathrm{CoM}} + {}^{\mathrm{W}}\mathcal{T}^{Z}_{\mathrm{Waist}} / \\ \mathcal{T}_{\mathrm{Posture}} \end{pmatrix} << \left(\mathcal{C}_{\substack{\mathrm{Joint} \\ \mathrm{Lims}}} + \mathcal{C}_{\substack{\mathrm{Vel.} \\ \mathrm{Lims}}}\right) \tag{7.21}$$

where ${}^{(A)}\mathcal{T}_{(B)}$ represents a Cartesian tasks of the frame B w.r.t. the frame A or expressed in joint space if A is not specified. The $/$ and $+$ symbols are used to impose *strict* or *soft* priority, respectively, among sets of tasks. The symbol $<<$ is used to enforce desired constraint in the task execution. During the motion, the SoT is modified on-line to adapt the robot behaviour in each transition phase: specifically, the parent link of a desired task can be selected or the whole task can be disabled, effectively removing it from the stack.

The contacts are detected using a simple algorithm that checks if a contact force $\boldsymbol{F_C}$ along the normal to the surface $n_{\boldsymbol{C}}$ exceeds a certain threshold, set to 15 N for the hands and 50 N for the feet. $\boldsymbol{F_C}$ is measured from the force/torque sensor mounted at the sole and estimated for the arm end-effector by exploiting the readings from the torque sensors at the arms joint level.

Finally, the joint level controller, composed of a decentralized impedance controller with a torque feed-forward term runs at 2 kHz.

7.6.3 Computation time

The computational load of the pipeline during the wall-plank task is negligible: the complexity of the NLP in the *Cartesian-space planner* is low, thanks to the choice of a continuous description of the environment and the simplified centroidal dynamics under quasi-static assumption. Similarly, the computational time of the RRT in the Joint-space planner is, on average, less than one second and, as highlighted in Section 7.4.3, the NLP for trajectory interpolation is solved in a fraction of the time required by the Joint-space planner. However, reducing the computation time as much as possible was not in the scope of this work, which aimed at devising a framework to perform complex physical interaction tasks and successfully implementing it on a real scenario. In future work, the enhancement of the framework's performances will be targeted, and a comparison with similar planning and control frameworks will be performed to assess its validity w.r.t. state of the art strategies.

7.6.4 Results

The sequence of stances composing the wall-plank task is depicted in Figure 7.4, while Figure 7.5 reports the values of the contact forces at the legs during the experiment, computed by the *Cartesian-space planner* and the *Interaction controller*. The *Interaction controller* runs in closed-loop, continuously distributing the forces on the supporting links: this generates discrepancies w.r.t. the planned forces provided by the *Cartesian-space planner*, since they are computed based on a simplified model (CS) *before* the actual motion. During the flight

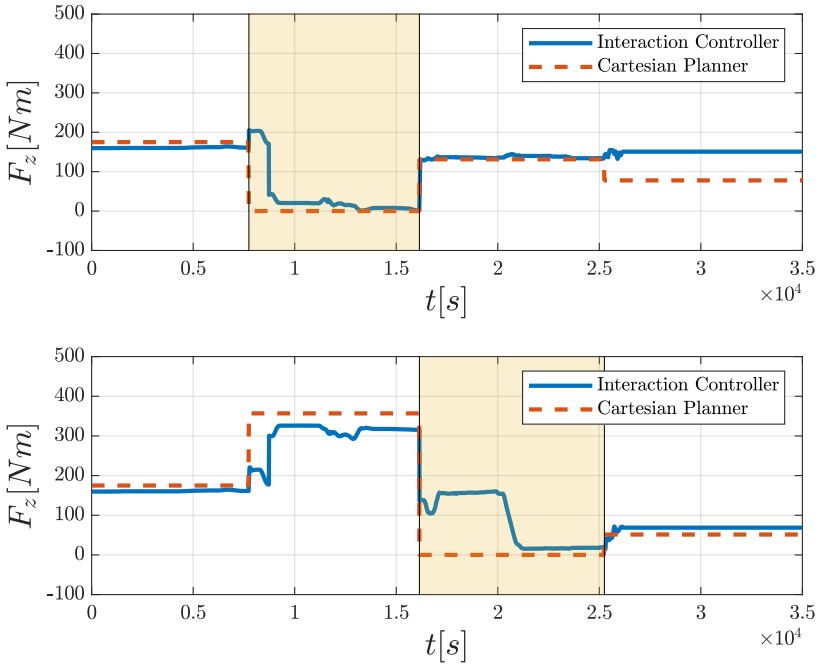


Figure 7.5 Values for the contact forces on the z-axis of left leg (above) and right leg (below) while performing Algorithm 1 for the *wall plank* task. The highlighted areas correspond to the flight phase of each leg.

phase a foot is lifted, and its relative contact force is set to zero. Videos of the experiments[1] show the hand-crafted motion and the one using the RRT trajectory planner. Notice also how the final configuration of the robot, in the wall plank pose, is robust to external pushes, thanks to the *Interaction controller*. The reactive balance granted by this control layer and the surface-reaching algorithm, which uses the force/torque sensors to detect impacts, provide robustness to modeling error of the surfaces and external disturbances. The velocity for each phase of the experiment can be increased, as long as the quasi-static stability is not violated.

7.7 Discussion

In this chapter a complete planning and control architecture is presented, that allows a humanoid robot to perform advanced real-world physical interaction tasks. The *Cartesian-space planner* uses a gradient-based optimization algorithm based on the CS to compute the contact forces for static equilibrium. The *Interaction controller* is responsible for the tracking of the references without violating the continuous balancing conditions imposed by

[1] https://www.youtube.com/playlist?list=PL7c1ZKncPan7g0tHDx5jP0kpt5si8G3pR

the CS and the Coulomb friction cones. A joint-space planner is used in combination with the *Cartesian-space planner*: while ad-hoc motion can be hand-crafted to reach the desired contacts, the presented algorithm guarantees general (self-)collision-free motions avoiding kinematics singularities. The choice fell on a global planning RRT algorithm to find a suitable trajectory, given the full-body model of the robot, and physical feasibility conditions, i.e. stability constraints (enforced using the CS) as well as environmental constraints, i.e. obstacle and contact constraints and a self-collision map. The realization of the aforementioned complex behaviour on a humanoid platform revealed the intrinsic limitations of its kinematic structure. The inherent poor flexibility of the robot, together with the lack of the torso pitch DoF increased the complexity of this particularly challenging task. The presented framework is capable to find a robust solution for this demanding task. Nevertheless, future enhancement will allow for a more organic pipeline towards physical autonomy. To reach the goal configuration each intermediate motion will be autonomously planned in terms of the number of contacts and body posture, without the need to manually outlining them at each phase of the sequence of motions. Since the joint-space planner randomly searches through the configuration space, a strategy to penalize burdensome and awkward movements will be conceived in favor of more natural-looking ones. The application of the presented pipeline to different scenarios is straightforward as long as they involve quasi-static motions and a continuous description of the environment can be provided: the task of wall plank will be extended to more articulated and dynamic motions, re-positioning the contacts to climb the wall with the feet or walk on the hands.

Chapter 8

Conclusions

This Ph.D. has focused on the design and the implementation of high-level control strategies on the humanoid robot COMAN+, addressing the challenging fields of biped locomotion and interaction with the environment. The novelty of the robot, built by the Humanoid & Human Centred Mechatronics Lab at Istituto Italiano di Tecnologia, presented additional obstacles that required the development of lower-level techniques necessaries for effective robot control. The resulting contribution of this thesis spanned strategies for humanoid motion concerning both lower-level control and higher behavioural level, splitting the work between the development of novel strategies and facing real-hardware difficulties in their implementation on a novel robotic platform.

8.1 Summary of contributions

The problem of high-level control of COMAN+ has been tackled by dividing it into two well-known categories, adding some hypotheses to simplify the problem, both on the environment and the properties of the motion. The main contribution concerning *gaited motions*, i.e. cyclic movements to locomote from one place to another, was the development of a lightweight **hybrid walking pattern generator** which generates feasible whole-body trajectories according to a few user-parameters. Drawing from the renowned preview control for biped locomotion and the virtual constraint theory, stemmed a novel walking pattern that allows flexible omnidirectional walking that only depends on a simple template model and a few step parameters. The forward and the lateral motions of the walking are addressed separately: the virtual constraints for the sagittal plane and the preview control based on equality-constrained quadratic problem for the lateral plane. The decoupled motions are merged in a consistent omnidirectional gait, allowing steering motions, lateral and backward stepping. The relevant

quantities for bipedal gaits of the resulting lower-dimensional system, i.e. the CoM and the feet trajectories, are merged in the lateral and sagittal direction in a feasible walking motion. The walking pattern generator is lighter in terms of computational complexity and parameter description compared to classical approaches, and a simple ROS interface can be used to issue on-line commands to the robot. Simulation experiments on COMAN+ demonstrated the effectiveness of the devised strategy: the robot walks and changes the nominal gait according to user-inputs, steering or moving backward without falling. Real-hardware experiments showed how the proposed walking pattern generator is promising, successfully allowing the robot to change its heading direction. While techniques for push recovery and disturbances rejection were tested separately, a stabilization layer was not incorporated in the walking pattern generator: since the prescribed trajectories are open-loop, the robot cannot react if its equilibrium gets disrupted by external disturbances or due to the accumulation of real hardware inaccuracies, losing balance after a few steps. Two **strategies for balancing** were explored, testing them in simulation and on the real robot: the first is a compliant stabilizer that adjusts the CoM of the robot in response to changes in the center of pressure, while the second is divided into an ankle and a torso stabilizer, that reacts against external forces in response to changes in the capture point.

Concerning *non-gaited motions*, i.e. specific actions aimed at the interaction with the environment (tasks), the contribution of this work was building a complete pipeline for **complex physical interaction tasks** and demonstrating its efficacy on the robot: a strong emphasis is put in planning feasible trajectories that satisfy joint limits and avoid self-collisions and singularities and realizing them on the robot without losing body balance. In this respect, the two-fold problem of whole-body motion and environment interaction needed to be tackled to achieve robust physical mobility and equilibrium. The effectiveness of the framework is tested on the humanoid COMAN+, fulfilling the designated goal: the robot successfully performs a challenging action for the average human being, the wall-plank, that consists in rising the feet on a wall and remaining suspended by distributing the weight on the four limbs. The motion highlights the critical role that contact interactions play in any motion that goes beyond nominal walking. Furthermore, it demonstrates how the proposed method can robustly produce highly non-linear trajectories in confined spaces, especially for a robot whose geometry heavily limits its dexterity, since it does not have the full flexibility and mobility of a human being. Ad-hoc motions can be hand-crafted to reach the desired contacts, the algorithm is generic: it guarantees (self-)collision-free, consistent motions that avoid kinematics singularities for any given task, as long as it involves quasi-static movements. Each layer is integrated into a framework that allows executing

complex motion actions while guaranteeing quasi-static equilibrium. The realization of the aforementioned complex behaviour on a humanoid platform revealed the intrinsic limitations of its kinematic structure: the inherent poor flexibility of the robot increased the complexity of this particularly challenging task. Nonetheless, the presented framework is capable to find a robust solution, and its application to different scenarios is straightforward as long as they involve quasi-static motions.

The main contribution concerning the *low-level control* was the design, implementation and testing of a **fail-safe impedance controller**: a controller capable of rejecting external perturbation while withstanding potential system failures without generating rapid motion reactions, which could compromise the hardware integrity or the safety of humans interacting with the robot. This is a fundamental prerequisite for a robotic platform used in real-time operations such as human-robot interactions or rescue missions, where the effects of failure should be reduced as much as possible to protect both the hardware and the human operators. The work was focused on the ankles of COMAN+: these joints are directly interacting with the environment and, consequently, they are subjected to high impact forces and disturbances. Additionally, the actuation system of the ankle of COMAN+ is based on a parallel mechanism driven by differential actuators, therefore direct impedance/torque control is not immediately available as for the other joints of the robot.

The controller was experimentally validated in the presence of disconnections: by simulating a hardware failure, it was proven how the ankle does not drift violently, but returns to its nominal position,

8.2 Future work

Humanoid research seeks to obtain dexterous and autonomous robots capable of mimicking the motion skills of a human being, involving locomotion, manipulation and, more in general, interaction with the environment. This thesis addressed these objectives dealing with the many issues arising from the implementation on a novel robotic platform. Successful results were achieved both concerning the locomotion and the interaction of the biped, as most of the strategies designed and implemented on COMAN+ showed positive results. Long-term goals envision the development of a range of gait adaptations and balancing controllers that will be mixed to provide a rich repertoire of locomotion skills. Successful strategies would grant the ability to move smoothly on uneven terrains and cluttered areas without any a priori knowledge of the surroundings. The robot should overcome obstacles without losing postural equilibrium while performing given tasks, exploiting intrinsic body balance and,

when necessary, objects or obstacles in the surroundings to improve its stability. Left to future work is then the relaxation of the set of assumptions selected to simplify the motion problems. Hypotheses can be made concerning the geometry and properties of the environment as well as the nature of the interactions of the robot with it: the more these limiting hypothesis are reduced, the more the gait can be made general, to fully exploit the potential of a modern, powerful torque-control robot. In particular for the walking pattern generator, hypotheses were made:

(i) flat terrain: the walking pattern generator will be extended for unstructured terrain, involving slopes and rough surfaces;

(ii) absence of external disturbances: external forces applied on the robot and generic disturbances will be considered to enhance the robustness of the controller;

(iii) non-cluttered environment: strategies to deal with obstacles will be devised, raising the autonomy of the gait.

Similarly, the physical interaction controller was conceived under some assumptions:

(iv) quasi-static motions: the full centroidal dynamics model will be used instead of the centroidal static equations, enabling more articulated and dynamic motions;

(v) a-priori knowledge of the environment: it will be possible to perform the physical interaction task in an unknown scenario

(vi) pre-computed contact sequence: the planner will autonomously generate a feasible sequence of manoeuvres to achieve the desired goal configuration.

Loosening first assumption (i) will require the walking pattern generator to allow more flexible gaits for walking on different terrain inclinations or walkable obstacles such as stairs. The integration on the whole body of the impedance controller developed for the ankles will grant active compliance against environment uncertainties (ii). Furthermore, successful rejection of external pushes will require the investigation of the stabilization techniques designed in this thesis, as well as the development of complex balancing controller approaches. This achievement will benefit both (ii) and (iv). The hypotheses (iii) and (v) will be tackled by incorporating the vision system of the robot inside the control architecture. Closing the loop on visual perception enables to overcome obstacles, perform tasks that are heavily dependent on the environment and greatly increase robot autonomy in general. Towards physical autonomy, the planning and control pipeline will generate each intermediate motion

to achieve the desired posture autonomously, in terms of the number of contacts and body posture (vi). Future work on the walking pattern generator will exploit fully the potential of the virtual constraints in the sagittal plane: the gait trajectories along this direction will be optimized to minimize power consumption and permit more dynamical movements. The *RoboGen* library will be populated with more methods to make it more flexible and intuitive for prototyping a novel robot or rapidly upload one or more elements to robots already included. Finally, future development will bridge *RoboGen* with the design software used to model the mechanical parts, to facilitate the transition between hardware components and their software counterparts.

8.3 Dissemination

The work presented in this thesis produced three main scientific publications, aimed at the development of control architectures and their deployment on the robotic platform COMAN+. The library *RoboGen* for the URDF-generation has been released.[1] The articles are listed in this section.

First-author articles

1. Article on the architecture of the fail-safe impedance controller on the ankles of COMAN+ (see Chapter 4). Presented at IROS 2018 and published on the relative conference proceedings.

   F. Ruscelli, A. Laurenzi, E. Mingo Hoffman and N. G. Tsagarakis, "A Fail-Safe Semi-Centralized Impedance Controller: Validation on a Parallel Kinematics Ankle," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 1-9.

2. Article on the hybrid walking pattern generator for omni-directional walking enabling on-line user commands for COMAN+ (see Chapter 5). Presented at IROS 2019 and published on the relative conference proceedings.

   F. Ruscelli, A. Laurenzi, E. M. Hoffman and N. G. Tsagarakis, "Synchronizing Virtual Constraints and Preview Controller: a Walking Pattern Generator for the Humanoid Robot COMAN+," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 3876-3881.

[1] https://advrhumanoids.github.io/robogen/

3. Article on the integration of a full pipeline for physical interaction tasks and a wall-plank demonstration with the robot COMAN+ (see Chapter 7. Presented at IROS 2020 and published on the relative conference proceedings.

   F. Ruscelli, M. P. Polverini, A. Laurenzi, E. M. Hoffman and N. G. Tsagarakis, "A Multi-Contact Motion Planning and Control Strategy for Physical Interaction Tasks Using a Humanoid Robot," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 25-29, 2020, Las Vegas, NV, USA (Virtual)

4. A paper has been submitted to the special issue on the research topic ”Towards Real-World Deployment of Legged Robots” of Frontiers in Robotics and AI. At the time of writing, the paper is in the process of being reviewed. This journal article aims at describing the full architecture of the omni-directional walking pattern generator, extending the previous paper on the hybrid walking patter generator:

   “Omnidirectional Walking Pattern Generator combining Virtual Constraints and Preview Control for Humanoid Robots”.

Other articles

Preliminary experiments on the planning and control architecture for physical interaction tasks were published in the journal paper:

M. Parigi Polverini, A. Laurenzi, E. Mingo Hoffman, F. Ruscelli, and N. G. Tsagarakis, “Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator,” IEEE Robotics and Automation Letters, presented at the IEEE International Conference on Robotics and Automation (ICRA), April 2020

References

Aftab, Z., Robert, T., and Wieber, P. (2012). Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 159–164.

Albu-Schäffer, A., Fischer, M., Schreiber, G., Schoeppe, F., and Hirzinger, G. (2004). Soft robotics: what cartesian stiffness can obtain with passively compliant, uncoupled joints? In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS).*, volume 4, pages 3295–3301. IEEE.

Albu-Schäffer, A. and Hirzinger, G. (2002). Cartesian impedance control techniques for torque controlled light-weight robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 657–663.

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36.

Aubin, J.-P., Bayen, A. M., and Saint-Pierre, P. (2011). *Viability theory: new directions*. Springer Science & Business Media.

Audren, H., Vaillant, J., Kheddar, A., Escande, A., Kaneko, K., and Yoshida, E. (2014). Model preview control in multi-contact motion-application to a humanoid robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4030–4035.

Baillieul, J., Hollerbach, J., and Brockett, R. (1984). Programming and control of kinematically redundant manipulators. In *The 23rd IEEE Conference on Decision and Control*, pages 768–774.

Caron, S. (2020). Biped stabilization by linear feedback of the variable-height inverted pendulum model. In *IEEE International Conference on Robotics and Automation*.

Catalano, M. G., Grioli, G., Farnioli, E., Serio, A., Piazza, C., and Bicchi, A. (2014). Adaptive synergies for the design and control of the pisa/iit softhand. *The International Journal of Robotics Research*, 33(5):768–782.

Chestnutt, J. E., Lau, M., Cheung, G. K. M., Kuffner, J., Hodgins, J. K., and Kanade, T. (2005). Footstep Planning for the Honda ASIMO Humanoid. In *IEEE International Conference on Robotics and Automation*, pages 629–634.

Chevallereau, C., Abba, G., Aoustin, Y., Plestan, F., Westervelt, E. R., Canudas-De-Wit, C., and Grizzle, J. W. (2003). Rabbit: a testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79.

Chevallereau, C., Grizzle, J. W., and Shih, C. (2010). Steering of a 3d bipedal robot with an underactuated ankle. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1242–1247.

Da, X., Harib, O., Hartley, R., Griffin, B., and Grizzle, J. W. (2016). From 2d design of underactuated bipedal gaits to 3d implementation: Walking with speed tracking. *IEEE Access*, 4:3469–3478.

Dai, H., Valenzuela, A., and Tedrake, R. (2014). Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302.

Dalibard, S., El Khoury, A., Lamiraux, F., Nakhaei, A., Taïx, M., and Laumond, J.-P. (2013). Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach. *The International Journal of Robotics Research*, 32(9-10):1089–1103.

Diedam, H., Dimitrov, D., Wieber, P., Mombaur, K., and Diehl, M. (2008). Online walking gait generation with adaptive foot positioning through linear model predictive control. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1121–1126.

Englsberger, J., Werner, A., Ott, C., Henze, B., Roa, M. A., Garofalo, G., Burger, R., Beyer, A., Eiberger, O., Schmid, K., and Albu-Schäffer, A. (2014). Overview of the torque-controlled humanoid robot toro. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 916–923.

Escande, A., Kheddar, A., and Miossec, S. (2013). Planning contact points for humanoid robots. *Robotics and Autonomous Systems*, 61(5):428–442.

Featherstone, R. (2010). A beginner's guide to 6-d vectors (part 1). *IEEE Robotics Automation Magazine*, 17(3):83–94.

Goswami, A. (1999). Postural stability of biped robots and the foot-rotation indicator (FRI) point. *International Journal of Robotics Research*, 18(6):523–533.

Griffin, R. J. and Leonessa, A. (2016). Model predictive control for dynamic footstep adjustment using the divergent component of motion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1763–1768.

Griffin, R. J., Wiedebach, G., Bertrand, S., Leonessa, A., and Pratt, J. (2017). Walking stabilization using step timing and location adjustment on the humanoid robot, atlas. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 667–673.

Grimes, J. A. and Hurst, J. W. (2012). The design of atrias 1.0 a unique monopod, hopping robot. In *Adaptive Mobile Robotics*, pages 548–554. World Scientific.

Grizzle, J. W., Chevallereau, C., Sinnet, R. W., and Ames, A. D. (2014). Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica*, 50(8):1955–1988.

Grizzle, J. W., Hurst, J., Morris, B., Park, H., and Sreenath, K. (2009). Mabel, a new robotic bipedal walker and runner. In *2009 American Control Conference*, pages 2030–2036.

Hashimoto, K., Sugahara, Y., Sunazuka, H., Tanaka, C., Ohta, A., Kawase, M., Hun-ok Lim, and Takanishi, A. (2006). Biped landing pattern modification method with nonlinear compliance control. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1213–1218.

Hauser, K., Bretl, T., Harada, K., and Latombe, J. C. (2008). Using motion primitives in probabilistic sample-based planning for humanoid robots. *Springer Tracts in Advanced Robotics*, 47:507–522.

Henze, B., Roa, M. A., and Ott, C. (2016). Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *The International Journal of Robotics Research*, 35(12):1522–1543.

Herdt, A., Diedam, H., Wieber, P.-B., Dimitrov, D., Mombaur, K., and Diehl, M. (2010a). Online walking motion generation with automatic foot step placement. *Advanced Robotics*, 24:719–737.

Herdt, A., Diedam, H., Wieber, P.-B., Dimitrov, D., Mombaur, K., and Diehl, M. (2010b). Online walking motion generation with automatic footstep placement. *Advanced Robotics*, 24(5-6):719–737.

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. (1998). The development of honda humanoid robot. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 2, pages 1321–1326 vol.2.

Hirose, M. and Ogawa, K. (2007). Honda humanoid robots development. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):11–19.

Hobbelen, D. G. E. and Wisse, M. (2007). A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm. *IEEE Transactions on Robotics*, 23(6):1213–1224.

Hun-ok Lim, Setiawan, S., and Takanishi, A. (2001). Balance and impedance control for biped humanoid robot locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*, volume 1, pages 494–499.

Hyon, S., Hale, J. G., and Cheng, G. (2007). Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Transactions on Robotics*, 23(5):884–898.

Jong Hyeon Park and Hoam Chung (1999). Hybrid control for biped robots using impedance control and computed-torque control. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1365–1370.

Kajita, S., Benallegue, M., Cisneros, R., Sakaguchi, T., Nakaoka, S., Morisawa, M., Kaneko, K., and Kanehiro, F. (2017). Biped walking pattern generation based on spatially quantized dynamics. *IEEE-RAS International Conference on Humanoid Robots*, pages 599–605.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003a). Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 2, pages 1620–1626 vol.2.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003b). Resolved momentum control: humanoid motion planning based on the linear and angular momentum. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1644–1650 vol.2.

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., and Hirukawa, H. (2001). The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 1, pages 239–246 vol.1.

Kajita, S., Morisawa, M., Harada, K., Kaneko, K., Kanehiro, F., Fujiwara, K., and Hirukawa, H. (2006). Biped walking pattern generator allowing auxiliary zmp control. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2993–2999.

Kajita, S., Morisawa, M., Miura, K., Nakaoka, S., Harada, K., Kaneko, K., Kanehiro, F., and Yokoi, K. (2010). Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4489–4496.

Kajita, S., Nagasaki, T., Kaneko, K., and Hirukawa, H. (2007). Zmp-based biped running control. *IEEE Robotics Automation Magazine*, 14(2):63–72.

Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., and Akachi, K. (2008). Humanoid robot hrp-3. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2471–2478.

Katayama, T., Ohki, T., Inoue, T., and Kato, T. (1985). Design of an optimal controller for a discrete-time system subject to previewable demand. *International Journal of Control*, 41(3):677–699.

Kato, I., Ohteru, S., Kobayashi, H., Shirai, K., and Uchiyama, A. (1974). Information-power machine with senses and limbs. In *On theory and practice of robots and manipulators*, pages 11–24. Springer.

Khadiv, M., Herzog, A., Moosavian, S. A. A., and Righetti, L. (2017). A robust walking controller based on online step location and duration optimization for bipedal locomotion. *arXiv preprint arXiv:1704.01271*.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53.

Kim, J., Lee, Y., Kwon, S., Seo, K., Kwak, H., Lee, H., and Roh, K. (2012). Development of the lower limbs for a humanoid robot. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4000–4005.

Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., and Tedrake, R. (2016). Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40(3):429–455.

Lanari, L., Hutchinson, S., and Marchionni, L. (2014). Boundedness issues in planning of locomotion trajectories for biped robots. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 951–958.

Laurenzi, A., Hoffman, E. M., Polverini, M. P., and Tsagarakis, N. G. (2018). Balancing Control through Post-Optimization of Contact Forces. In *IEEE/RAS International Conference of Humanoid Robots*, Beijing, China. IEEE/RAS.

Laurenzi, A., Mingo Hoffman, E., Muratore, L., and Tsagarakis, N. G. (2019). CartesI/O: A ROS Based Real-Time Capable Cartesian Control Framework. In *IEEE International Conference on Robotics and Automation*.

Lee, S. and Goswami, A. (2007). Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4667–4672.

Lee, S. and Goswami, A. (2010). Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3157–3162.

Li, Z., Tsagarakis, N., and Caldwell, D. G. (2012). A Passivity Based Admittance Control for Stabilizing the Compliant Humanoid COMAN. In *IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, Osaka, Japan.

Li, Z., Tsagarakis, N. G., and Caldwell, D. G. (2012a). A passivity based admittance control for stabilizing the compliant humanoid coman. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 43–49.

Li, Z., Vanderborght, B., Tsagarakis, N. G., Colasanto, L., and Caldwell, D. G. (2012b). Stabilization for the compliant humanoid robot coman exploiting intrinsic and controlled compliance. In *2012 IEEE International Conference on Robotics and Automation*, pages 2000–2006.

Maciejewski, A. A. and Klein, C. A. (1989). The singular value decomposition: Computation and applications to robotics. *The International Journal of Robotics Research*, 8(6):63–79.

Manchester, I. R., Mettin, U., Iida, F., and Tedrake, R. (2011). Stable Dynamic Walking over Rough Terrain. *Robotics Research:The 14th International Symposium*, pages 123–138.

Mingo Hoffman, E., Perrin, N., G. Tsagarakis, N., and G. Caldwell, D. (2013). Upper limb compliant strategy exploiting external physical constraints for humanoid fall avoidance. In *IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, pages 397–402.

Mordatch, I., Todorov, E., and Popović, Z. (2012). Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):1–8.

Morisawa, M., Harada, K., Kajita, S., Kaneko, K., Sola, J., Yoshida, E., Mansard, N., Yokoi, K., and Laumond, J. (2009). Reactive stepping to prevent falling for humanoids. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 528–534.

Motahar, M. S., Veer, S., and Poulakakis, I. (2017). Steering a 3d limit-cycle walker for collaboration with a leader. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5251–5256.

Muratore, L., Laurenzi, A., Hoffman, E. M., Rocchi, A., Caldwell, D. G., and Tsagarakis, N. G. (2017). XBotCore: A Real-Time Cross-Robot Software Platform. In *First IEEE International Conference on Robotic Computing (IRC)*, pages 77–80. IEEE.

Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15.

Nashner, L. M. and McCollum, G. (1985). The organization of human postural movements: a formal basis and experimental synthesis. *Behavioral and brain sciences*, 8(1):135–150.

Nishi, T. and Sugihara, T. (2014). Motion planning of a humanoid robot in a complex environment using rrt and spatiotemporal post-processing techniques. *International Journal of Humanoid Robotics*, 11.

Nishiwaki, K. and Kagami, S. (2006). High frequency walking pattern generation based on preview control of zmp. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2667–2672.

Nishiwaki, K. and Kagami, S. (2009). Online walking control system for humanoids with short cycle pattern generation. *The International Journal of Robotics Research*, 28(6):729–742.

Ott, C., Roa, M. A., and Hirzinger, G. (2011). Posture and balance control for biped robots based on contact force optimization. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 26–33.

Pan, J., Chitta, S., and Manocha, D. (2012). Fcl: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation*, pages 3859–3866.

Papa, E. and Cappozzo, A. (1999). A telescopic inverted-pendulum model of the musculoskeletal system and its use for the analysis of the sit-to-stand motor task. *Journal of Biomechanics*, 32(11):1205–1212.

Parigi Polverini, M., Laurenzi, A., Mingo Hoffman, E., Ruscelli, F., and Tsagarakis, N. G. (2020). Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator. *IEEE Robotics and Automation Letters, presented at the IEEE International Conference on Robotics and Automation (ICRA)*, 5(2):859–866.

Parmiggiani, A., Maggiali, M., Natale, L., Nori, F., Schmitz, A., Tsagarakis, N., Victor, J. S., Becchi, F., Sandini, G., and Metta, G. (2012). The design of the icub humanoid robot. *International journal of humanoid robotics*, 9(04):1250027.

Posa, M., Kuindersma, S., and Tedrake, R. (2016). Optimization and stabilization of trajectories for constrained dynamical systems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1366–1373.

Pratt, G. and Williamson, M. (1995). Series elastic actuators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 399–406.

Pratt, J., Carff, J., Drakunov, S., and Goswami, A. (2006). Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207.

Pratt, J. E., Chew, C.-M., Torres, A., Dilworth, P., and Pratt, G. A. (2001). Virtual Model Control: An Intuitive Approach for Bipedal Locomotion. *I. J. Robotic Res.*, 20(2):129–143.

Raibert, M. H. (1986). *Legged robots that balance*. MIT press.

Razavi, H., Bloch, A. M., Chevallereau, C., and Grizzle, J. W. (2015). Restricted discrete invariance and self-synchronization for stable walking of bipedal robots. *Proceedings of the American Control Conference*, pages 4818–4824.

Reher, J., Cousineau, E. A., Hereid, A., Hubicki, C. M., and Ames, A. D. (2016). Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS. In *IEEE International Conference on Robotics and Automation*, pages 1794–1801.

Rezazadeh, S., Hubicki, C., Jones, M., Peekema, A., Van Why, J., Abate, A., and Hurst, J. (2015). Spring-mass walking with atrias in 3d: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot. In *ASME Dynamic Systems and Control Conference*.

Runge, C., Shupert, C., Horak, F., and Zajac, F. (1999). Ankle and hip postural strategies defined by joint torques. *Gait & posture*, 10(2):161–170.

Saab, L., Ramos, O. E., Keith, F., Mansard, N., Soueres, P., and Fourquet, J.-Y. (2013). Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints. *IEEE Transactions on Robotics*, 29(2):346–362.

Sardain, P. and Bessonnet, G. (2004). Zero moment point - Measurements from a human walker wearing robot feet as shoes. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 34(5):638–648.

Scianca, N., Cognetti, M., De Simone, D., Lanari, L., and Oriolo, G. (2016). Intrinsically stable MPC for humanoid gait generation. *IEEE-RAS International Conference on Humanoid Robots*, pages 601–606.

Shimmyo, S., Sato, T., and Ohnishi, K. (2013). Biped walking pattern generation by using preview control based on three-mass model. *IEEE Transactions on Industrial Electronics*, 60(11):5137–5147.

Stasse, O., Flayols, T., Budhiraja, R., Giraud-Esclasse, K., Carpentier, J., Mirabel, J., Del Prete, A., Souères, P., Mansard, N., Lamiraux, F., Laumond, J.-P., Marchionni, L., Tome, H., and Ferro, F. (2017). TALOS: A new humanoid research platform targeted for industrial applications. In *International Conference on Humanoid Robotics, ICHR, Birmingham 2017*, IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids),, Birmingham, United Kingdom. IEEE.

Stephens, B. (2007). Integral control of humanoid balance. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4020–4027.

Stephens, B. J. and Atkeson, C. G. (2010). Push recovery by stepping for humanoid robots with force controlled joints. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 52–59.

Şucan, I. A., Moll, M., and Kavraki, L. E. (2012). The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82. http://ompl.kavrakilab.org.

Tassa, Y., Erez, T., and Todorov, E. (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913.

Tellez, R., Ferro, F., Garcia, S., Gomez, E., Jorge, E., Mora, D., Pinyol, D., Oliver, J., Torres, O., Velazquez, J., and Faconti, D. (2008). Reem-b: An autonomous lightweight human-size humanoid robot. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 462–468.

Tsagarakis, N. G., Caldwell, D. G., Negrello, F., Choi, W., Baccelliere, L., Loc, V., Noorden, J., Muratore, L., Margan, A., Cardellino, A., Natale, L., Mingo Hoffman, E., Dallali, H., Kashiri, N., Malzahn, J., Lee, J., Kryczka, P., Kanoulas, D., Garabini, M., Catalano, M., Ferrati, M., Varricchio, V., Pallottino, L., Pavan, C., Bicchi, A., Settimi, A., Rocchi, A., and Ajoudani, A. (2017). WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments. *Journal of Field Robotics*, 34(7):1225–1259.

Tsagarakis, N. G., Morfey, S., Medrano Cerda, G., Zhibin, L., and Caldwell, D. G. (2013). Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control. In *2013 IEEE International Conference on Robotics and Automation*, pages 673–678.

Tsagarakis, N. G., Zhibin Li, Saglia, J., and Caldwell, D. G. (2011). The design of the lower body of the compliant humanoid robot "ccub". In *2011 IEEE International Conference on Robotics and Automation*, pages 2035–2040.

Urata, J., Nshiwaki, K., Nakanishi, Y., Okada, K., Kagami, S., and Inaba, M. (2011). Online decision of foot placement using singular lq preview regulation. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 13–18.

Vaillant, J., Kheddar, A., Audren, H., Keith, F., Brossette, S., Escande, A., Bouyarmane, K., Kaneko, K., Morisawa, M., Gergondet, P., Yoshida, E., Kajita, S., and Kanehiro, F. (2016). Multi-contact vertical ladder climbing with an hrp-2 humanoid. *Autonomous Robots*, 40.

Verrelst, B., Stasse, O., Yokoi, K., and Vanderborght, B. (2006). Dynamically stepping over obstacles by the humanoid robot hrp-2. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 117–123.

Vukobratovic, M. and Juricic, D. (1969). Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*, BME-16(1):1–6.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.

Walker, I. D. and Marcus, S. I. (1988). Subtask performance by redundancy resolution for redundant robot manipulators. *IEEE Journal on Robotics and Automation*, 4(3):350–354.

Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):93–101.

Werner, A., Henze, B., Rodriguez, D. A., Gabaret, J., Porges, O., and Roa, M. A. (2016). Multi-contact planning and control for a torque-controlled humanoid robot. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5708–5715. IEEE.

Westervelt, E., Chevallereau, C., Morris, B., Grizzle, J., and Ho Choi, J. (2007). *Feedback Control of Dynamic Bipedal Robot Locomotion*, volume 26 of *Automation and Control Engineering*. CRC Press.

Wieber, P.-B. (2006). Holonomy and nonholonomy in the dynamics of articulated motion. In *Fast motions in biomechanics and robotics*, pages 411–425. Springer.

Zhou, C., Li, Z., Castano, J., Dallali, H., Tsagarakis, N. G., and Caldwell, D. G. (2014). A passivity based compliance stabilizer for humanoid robots. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1487–1492.

Zhou, C. and Tsagarakis, N. (2018). On the comprehensive kinematics analysis of a humanoid parallel ankle mechanism. *Journal of Mechanisms and Robotics*, 10(5).